Abstract

This article provides a short proof of the Prime Number Theorem in several equivalent forms, most notably $\pi(x) \sim x / \ln x$ where $\pi(x)$ is the number of primes no larger than $x$. It also defines other basic number-theoretic functions related to primes like Chebyshev’s $\vartheta$ and $\psi$ and the “$n$-th prime number” function $p_n$. We also show various bounds and relationship between these functions are shown. Lastly, we derive Mertens’ First and Second Theorem, i.e. $\sum_{p \leq x} \frac{\ln p}{p} = \ln x + O(1)$ and $\sum_{p \leq x} \frac{1}{p} = \ln \ln x + M + O(1/\ln x)$. We also give explicit bounds for the remainder terms.

The proof of the Prime Number Theorem builds on a library of Dirichlet series and analytic combinatorics. We essentially follow the presentation by Newman [6]. The core part of the proof is a Tauberian theorem for Dirichlet series, which is proven using complex analysis and then used to strengthen Mertens’ First Theorem to $\sum_{p \leq x} \frac{\ln p}{p} = \ln x + c + o(1)$.

A variant of this proof has been formalised before by Harrison in HOL Light [5], and formalisations of Selberg’s elementary proof exist both by Avigd et al. [2] in Isabelle and by Carneiro [3] in Metamath. The advantage of the analytic proof is that, while it requires more powerful mathematical tools, it is considerably shorter and clearer. This article attempts to provide a short and clear formalisation of all components of that proof using the full range of mathematical machinery available in Isabelle, staying as close as possible to Newman’s simple paper proof.
## Contents

1 Auxiliary material .............................................. 3

2 Ingham’s Tauberian Theorem .................................. 47

3 Prime-Counting Functions .................................... 61
   3.1 Definitions .............................................. 62
   3.2 Basic properties ....................................... 64
   3.3 The \( n \)-th prime number ............................ 66
   3.4 Relations between different prime-counting functions .... 70
   3.5 Bounds .................................................. 73
   3.6 The asymptotic form of Mertens’ First Theorem ......... 81

4 The Prime Number Theorem .................................. 86
   4.1 Constructing Newman’s function ....................... 86
   4.2 The asymptotic expansion of \( \mathfrak{M} \) .............. 95
   4.3 The asymptotics of the prime-counting functions ....... 101

5 Mertens’ Theorems .............................................. 105
   5.1 Absolute Bounds for Mertens’ First Theorem .......... 106
   5.2 Mertens’ Second Theorem ............................... 116
1 Auxiliary material

theory Prime-Number-Theorem-Library
imports
  Zeta-Function.Zeta-Function
  HOL.Real-Asymp.Real-Asymp
begin

lemma real-root-decreasing: \(0 < n \implies n \leq N \implies 1 \leq x \implies \sqrt[n]{x} \leq \sqrt[N]{x}\)
by (auto simp add: order-le-less real-root-strict-decreasing)

lemma real-root-increasing: \(0 < n \implies n \leq N \implies 0 \leq x \implies x \leq 1 \implies \sqrt[n]{x} \leq \sqrt[N]{x}\)
by (auto simp add: order-le-less real-root-strict-increasing)

lemma frontier-real-Ici [simp]:
  fixes a :: real
  shows frontier {a..} = {a}
unfolding frontier-def by (auto simp: interior-real-semiline)

lemma sum-upto-ln-conv-sum-upto-mangoldt:
  sum-upto (\(\lambda n. \ln (\text{real } n)\)) x = sum-upto (\(\lambda n. \text{mangoldt } n \cdot \text{nat } \lfloor x / \text{real } n \rfloor\)) x
proof
  have sum-upto (\(\lambda n. \ln (\text{real } n)\)) x = sum-upto (\(\lambda n. \sum d \mid d \text{ dvd } n. \text{mangoldt } d\)) x
    by (intro sum-upto-cong) (simp-all add: mangoldt-sum)
  also have \(\ldots = \text{sum-upto } (\lambda k. \text{sum-upto } (\lambda d. \text{mangoldt } k) (x / \text{real } k))\) x
    by (rule sum-upto-sum-divisors)
  also have \(\ldots = \text{sum-upto } (\lambda n. \text{mangoldt } n \cdot \text{nat } \lfloor x / \text{real } n \rfloor)\) x
    unfolding sum-upto-altdef by (simp add: mult-ac)
  finally show ?thesis.
qed

lemma ln-fact-conv-sum-upto-mangoldt:
  \(\ln (\text{fact } n) = \text{sum-upto } (\lambda k. \text{mangoldt } k \cdot (n \text{ div } k))\) n
proof
  have simp: \(\{0<..\text{Suc } n\} = \text{insert } \text{Suc } n \{0<..n\}\) for n by auto
  have ln (\(\text{fact } n\)) = sum-upto (\(\lambda n. \ln (\text{real } n)\)) n
    by (induction n) (auto simp: sum-upto-altdef nat-add-distrib ln-mul)
  also have \(\ldots = \text{sum-upto } (\lambda k. \text{mangoldt } k \cdot (n \text{ div } k))\) n
    unfolding sum-upto-ln-conv-sum-upto-mangoldt
    by (intro sum-upto-cong) (auto simp: floor-divide-of-nat-eq)
  finally show ?thesis.
qed

lemma powr-sum: \(x \neq 0 \implies \text{finite } A \implies x \text{ powr sum } f A = (\prod y \in A. x \text{ powr } f y)\)
by (simp add: powr-def exp-sum sum-distrib-right)

3
lemma fds-abs-converges-comparison-test:
  fixes s :: 'a :: dirichlet-series
  assumes eventually (λn. norm (fds-nth f n) ≤ fds-nth g n) at-top and fds-converges g (s · 1)
  shows fds-abs-converges f s
  unfolding fds-abs-converges-def
proof (rule summable-comparison-test-ev)
  from assms(2) show summable (λn. fds-nth g n / n powr (s · 1))
    by (auto simp: fds-converges-def)
  from assms(1) eventually-gt-at-top[of 0]
    show eventually (λn. norm (norm (fds-nth f n / nat-power n s)) ≤
      fds-nth g n / real n powr (s · 1)) at-top
    by eventually-elim (auto simp: norm-divide norm-nat-power intro: divide-right-mono)
qed

lemma fds-converges-scaleR [intro]:
  assumes fds-converges f s
  shows fds-converges (c *ₗ f) s
proof –
  from assms have summable (λn. c *ₗ (fds-nth f n / nat-power n s))
    by (intro summable-scaleR-right) (auto simp: fds-converges-def)
  also have (λn. c *ₗ (fds-nth f n / nat-power n s)) = (λn. (c *ₗ fds-nth f n / nat-power n s))
    by (simp add: scaleR-conv-of-real)
  finally show thesis by (simp add: fds-converges-def)
qed

lemma fds-abs-converges-scaleR [intro]:
  assumes fds-abs-converges f s
  shows fds-abs-converges (c *ₗ f) s
proof –
  from assms have summable (λn. abs c * norm (fds-nth f n / nat-power n s))
    by (intro summable-scaleR-right) (auto simp: fds-abs-converges-def)
  also have (λn. abs c * norm (fds-nth f n / nat-power n s)) =
    (λn. norm (c *ₗ fds-nth f n / nat-power n s)) by (simp add: norm-divide)
  finally show thesis by (simp add: fds-abs-converges-def)
qed

lemma conv-abcissa-scaleR: conv-abcissa (scaleR c f) ≤ conv-abcissa f
  by (rule conv-abcissa-mono) auto

lemma abs-conv-abcissa-scaleR: abs-conv-abcissa (scaleR c f) ≤ abs-conv-abcissa f
  by (rule abs-conv-abcissa-mono) auto

lemma fds-converges-mult-const-left [intro]:
  fds-converges f s ⇒ fds-converges (fds-const c * f) s
by (auto simp: fds-converges-def dest: summable-mult[of - c])

lemma fds-abs-converges-mult-const-left [intro]:
  fds-abs-converges f s \implies fds-abs-converges (fds-const c * f) s
by (auto simp: fds-abs-converges-def norm-mult norm-divide dest: summable-mult[of - norm c])

lemma conv-abscissa-mult-const-left:
  conv-abscissa (fds-const c * f) \leq conv-abscissa f
by (intro conv-abscissa-mono) auto

lemma abs-conv-abscissa-mult-const-left:
  abs-conv-abscissa (fds-const c * f) \leq abs-conv-abscissa f
by (intro abs-conv-abscissa-mono) auto

lemma fds-converges-mult-const-right [intro]:
  fds-converges f s \implies fds-converges (f * fds-const c) s
by (auto simp: fds-converges-def dest: summable-mult2[of - c])

lemma fds-abs-converges-mult-const-right [intro]:
  fds-abs-converges f s \implies fds-abs-converges (f * fds-const c) s
by (auto simp: fds-abs-converges-def norm-mult norm-divide dest: summable-mult2[of - norm c])

lemma conv-abscissa-mult-const-right:
  conv-abscissa (f * fds-const c) \leq conv-abscissa f
by (intro conv-abscissa-mono) auto

lemma abs-conv-abscissa-mult-const-right:
  abs-conv-abscissa (f * fds-const c) \leq abs-conv-abscissa f
by (intro abs-conv-abscissa-mono) auto

lemma bounded-coeffs-imp-fds-abs-converges:
  fixes s :: 'a :: dirichlet-series and f :: 'a fds
  assumes Bseq (fds-nth f) s * 1 > 1
  shows fds-abs-converges f s
proof
  from assms obtain C where C: \forall n. norm (fds-nth f n) \leq C
  by (auto simp: Bseq-def)
  show ?thesis
  proof (rule fds-abs-converges-comparison-test)
    from (s * 1 > 1) show fds-converges (C *R fds-zeta) (s * 1)
    by (intro fds-abs-converges-imp-converges) auto
    from C show eventually (\lambda n. norm (fds-nth f n) \leq fds-nth (C *R fds-zeta) n) at-top
    by (intro always-eventually) (auto simp: fds-nth-zeta)
  qed
  qed
lemma bounded-coeffs-imp-fds-abs-converges':
 fixes s :: 'a :: dirichlet-series and f :: 'a fds
 assumes Bseq (λn. fds-nth f n * nat-power n s0) s · 1 > 1 - s0 · 1
 shows fds-abs-converges f s
 proof
 have fds-nth (fds-shift s0 f) = (λn. fds-nth f n * nat-power n s0)
   by (auto simp: fun-eq_iff)
 with assms have Bseq (fds-nth (fds-shift s0 f)) by simp
 with assms(2) have fds-abs-converges (fds-shift s0 f) (s + s0)
   by (intro bounded-coeffs-imp-fds-abs-converges) (auto simp: algebra-simps)
 thus ?thesis by simp
 qed

lemma bounded-coeffs-imp-abs-conv-abscissa-le:
 fixes s :: 'a :: dirichlet-series and f :: 'a fds and c :: ereal
 assumes Bseq (λn. fds-nth f n * nat-power n s) 1 - s · 1 ≤ c
 shows abs-conv-abscissa f ≤ c
 proof (rule abs-conv-abscissa-leI-weak)
   fix x assume c < ereal x
   have ereal (1 - s · 1) ≤ c by fact
   also have ... < ereal x by fact
   finally have 1 - s · 1 < ereal x by simp
   thus fds-abs-converges f (of-real x)
     by (intro bounded-coeffs-imp-fds-abs-converges'[OF assms(1)]) auto
 qed

lemma bounded-coeffs-imp-abs-conv-abscissa-le-1:
 fixes s :: 'a :: dirichlet-series and f :: 'a fds
 assumes Bseq (λn. fds-nth f n)
 shows abs-conv-abscissa f ≤ 1
 proof
 have [simp]: fds-nth f n * nat-power n 0 = fds-nth f n for n
   by (cases n = 0) auto
 show ?thesis
   by (rule bounded-coeffs-imp-abs-conv-abscissa-le[where s = 0]) (insert assms, auto simp:)
 qed

lemma
 fixes a b c :: real
 assumes ab: a + b > 0 and c: c < -1
 shows set-integrable-powr-at-top: (λx. (b + x) powr c) absolutely-integrable-on {a<..}
   and set-lebesgue-integral-powr-at-top:
   (f x ∈ {a<..}. (b + x) powr c) ∂lborel = -((b + a) powr (c + 1) / (c + 1))
and pour-has-integral-at-top:

\((\lambda x. (b + x) \cdot c) \text{ has-integral } -( (b + a) \cdot \frac{(c + 1)}{(c + 1)} ))\)

\{a<..\}

proof —

let \(\alpha = \lambda x. (b + x) \cdot c\) and \(\beta = \lambda x. (b + x) \cdot \frac{(c + 1)}{(c + 1)}\)

have limits: \((\langle \alpha \circ \text{real-of-ereal} \rangle \longrightarrow \alpha a)\) (at-right (ereal a))

\((\langle \beta \circ \text{real-of-ereal} \rangle \longrightarrow \beta)\) (at-left \(\infty\))

using \(c\ ab\ unfolding\ \text{ereal-tends-to-simps1}\ by\ \text{(real-asym}\ \text{simp: field-simps)} +

have 1: set-integrable \(\text{borel}\ \langle\text{einterval}\ a\ \infty\rangle\ \beta\ using\ ab\ c\ limits\)

by (intro interval-integral-FTC-nonneg) (auto intro!: derivative-eq-intros)

thus 2: \(\alpha\ \text{absolutely-integrable-on} \{a<..\}\)

by (auto simp: set-integrable-def integrable-completion)

have \(\text{LBINT}\ x=\text{ereal}\ a..\infty. (b + x) \cdot c = 0 - \beta a\ using\ ab\ c\ limits\)

by (intro interval-integral-FTC-nonneg) (auto intro!: derivative-eq-intros)

thus 3: \(\{\{ f \in \{a<..\}, ((b + x) \cdot c) \in \partial\text{borel}\} = -( (b + a) \cdot \frac{(c + 1)}{(c + 1)} )\}\)

by (simp add: interval-integral-to-infinity-eq)

show \((\langle \alpha \text{ has-integral } -( (b + a) \cdot \frac{(c + 1)}{(c + 1)} )\rangle \{a<..\}\)

using set-borel-integral-eq-integral[OF 1] 3 by (simp add: has-integral-iff)

qed

lemma fds-converges-altdef2:

\(\text{fds-converges}\ f\ s \longleftrightarrow\ \text{convergent}\ (\lambda N. \text{eval-fds}\ (\text{fds-truncate}\ N\ f)\ s)\)

\(\text{unfolding\ fds-converges-def\ summable-iff-convergent'}\ \text{eval-fds-truncate}\)

by (auto simp: not-le intro!: convergent-cong always-eventually sum.mono-neutral-right)

lemma tendsto-eval-fds-truncate:

assumes \(\text{fds-converges}\ f\ s\)

shows \(\lambda N. \text{eval-fds}\ (\text{fds-truncate}\ N\ f)\ s\longrightarrow\ \text{eval-fds}\ f\ s\)

proof —

have \(\lambda N. \text{eval-fds}\ (\text{fds-truncate}\ N\ f)\ s\longrightarrow\ \text{eval-fds}\ f\ s \longleftrightarrow\)

\(\lambda N. \sum_{i\in\{N\}} \text{fds-nth}\ f\ i\ /\ \text{nat-power}\ i\ s\longrightarrow\ \text{eval-fds}\ f\ s\)

unfolding eval-fds-truncate

by (intro filterlim-cong always-eventually allI sum.mono-neutral-left) (auto simp: not-le)

also have \ldots using assms

by (simp add: fds-converges-iff sums-def' atLeast0AtMost)

finally show \(?\text{thesis} .\)

qed

lemma linepath-translate-left: linepath \((c + a)\) \((c + a)\) = (\(\lambda x. c + a\) \(\circ\) linepath \(a\ b\))

by (auto simp: fun-eq-iff linepath-def algebra-simps)

lemma linepath-translate-right: linepath \((a + c)\) \((b + c)\) = (\(\lambda x. x + c\) \(\circ\) linepath \(a\ b\))

by (auto simp: fun-eq-iff linepath-def algebra-simps)

lemma integrable-on-affinity:
assumes $m \neq 0$ f integrable-on (cbox a b)
shows $\langle \lambda x. (1 / m) *R x - ((1 / m) *R c) \rangle \cdot$ cbox a b)

proof
from assms obtain I where $f$ has-integral I (cbox a b)
by (auto simp: integrable-on-def)
from has-integral-affinity[OF this assms, of c] show thesis
by (auto simp: integrable-on-def)

defined c

lemma has-integral-cmul-iff:
assumes $c \neq 0$
shows $\langle \lambda x. c *R f x \rangle$ has-integral (c *R I) A $\leftrightarrow$ (f has-integral I) A
using assms has-integral-cmul[of f I A c]
has-integral-cmul[of $\lambda x. c *R f x c *R I A$ inverse c] by (auto simp: field-simps)

lemma has-integral-affinity':
fixes a :: 'a::euclidean-space
assumes $f$ has-integral i (cbox a b) and $m > 0$
shows $\langle \lambda x. (m *R x + c) \rangle$ has-integral (i /R m $\cdot$ DIM('a))
(cbox ((a - c) /R m) ((b - c) /R m))

next

lemma has-integral-affinity-iff:
fixes f :: 'a::euclidean-space $\Rightarrow$ 'b :: real-normed-vector
assumes $m > 0$
shows $\langle \lambda x. ((1 / m) *R x + c) \rangle$ has-integral (1 /R m $\cdot$ DIM('a))
(cbox ((a - c) /R m) ((b - c) /R m)) $\leftrightarrow$
proof
  assume ?lhs
  from has-integral-affinity"[OF this, of I / m -c / R m] and (m > 0)
  show ?rhs by (simp add: field-simps vector-add-divide-simps)
next
  assume ?rhs
  from has-integral-affinity"[OF this, of m c]
  show ?lhs by simp
qed

lemma has-contour-integral-linepath-Reals-iff:
  fixes a b :: complex and f :: complex ⇒ complex
  assumes a ∈ Reals b ∈ Reals Re a < Re b
  shows (f has-contour-integral I) (linepath a b) \iff
  ((λx. f (of-real x)) has-integral I) {Re a..Re b}
proof
  from assms have [simp]: of-real (Re a) = a of-real (Re b) = b
    by (simp-all add: complex-eq-iff)
  from assms have a ≠ b by auto
  have ((λx. f (of-real x)) has-integral I) (cbox a b)
    \iff
    ((λx. f (a + b * of-real x - a * of-real x)) has-integral I / R (Re b - Re a)) {0..1}
    by (subst has-integral-affinity-iff [of Re b - Re a - Re a, symmetric])
    (insert assms, simp-all add: field-simps scaleR-conv-of-real)
  also have (λx. f (a + b * of-real x - a * of-real x)) =
    (λx. f (a + b * of-real x - a * of-real x) * (b - a)) / R (Re b - Re a)
    using [a ≠ b] by (auto simp: field-simps fun-eq-iff scaleR-conv-of-real)
  also have \ldots \iff (f has-contour-integral I) (linepath a b)
    unfolding has-contour-integral-def
    by (intro has-integral-cong) (simp add: vector-derivative-linepath-within)
  finally show ?thesis by simp
qed

lemma contour-integrable-linepath-Reals-iff:
  fixes a b :: complex and f :: complex ⇒ complex
  assumes a ∈ Reals b ∈ Reals Re a < Re b
  shows (f contour-integrable-on linepath a b) \iff
    ((λx. f (of-real x)) integrable-on {Re a..Re b})
  using has-contour-integral-linepath-Reals-iff[OF assms, of f]
  by (auto simp: contour-integrable-on-def integrable-on-def)

lemma contour-integral-linepath-Reals-eq:
  fixes a b :: complex and f :: complex ⇒ complex
  assumes a ∈ Reals b ∈ Reals Re a < Re b
shows \( \text{contour-integral (linepath a b)} f = \text{integral} \{ \text{Re a..Re b} \} (\lambda x. f \text{ (of-real } x)) \)

proof (cases \( f \text{ contour-integrable-on linepath a b} \))

  case True
  thus \( \text{?thesis using has-contour-integral-linepath-Reals-iff [OF assms, of f]} \)
    using has-contour-integral-integral has-contour-integral-unique by blast
  next
  case False
  thus \( \text{?thesis using contour-integrable-linepath-Reals-iff [OF assms, of f]} \)
    by (simp add: not-integrable-contour-integral not-integrable-integral)
qed

lemma has-contour-integral-linepath-same-Im-iff:
  fixes \( a \text{ b :: complex and } f :: \text{complex} \Rightarrow \text{complex} \)
  assumes \( \text{Im a = Im b Re a < Re b} \)
  shows \( (f \text{ has-contour-integral I}) \text{ (linepath a b)} \longleftrightarrow \)
    \( ((\lambda x. f \text{ (of-real } x + \text{Im a} * i)) \text{ has-integral I}) \text{ (Re a..Re b)} \)

proof –
  have deriv: \( \text{vector-derivative ((}\lambda x. x - \text{Im a} * i) \circ \text{linepath a b})\text{ (at y)} = b - a \)
    for \( y \)
    using linepath-translate-right[of \( a - \text{Im a} * i \text{ b, symmetric} \) by simp
  have \( (f \text{ has-contour-integral I}) \text{ (linepath a b)} \longleftrightarrow \)
    \( ((\lambda x. f \text{ (x + Im a * i)}) \text{ has-contour-integral I}) \text{ (linepath } (a - \text{Im a} * i) \text{ (a \text{ Im a} * i)} \text{ (b \text{ Im a} * i)} \)
    \text{ using linepath-translate-right[of a - Im a * i b] deriv by (simp add: has-contour-integral)}
  also have \( \ldots \longleftrightarrow ((\lambda x. f \text{ (x + Im a * i)}) \text{ has-integral I}) \text{ (Re a..Re b)} \text{ using assms} \)
    by (subst has-contour-integral-linepath-Reals-iff) (auto simp: complex-is-Real-iff)
  finally show \( \text{?thesis} \).
qed

lemma contour-integrable-linepath-same-Im-iff:
  fixes \( a \text{ b :: complex and } f :: \text{complex} \Rightarrow \text{complex} \)
  assumes \( \text{Im a = Im b Re a < Re b} \)
  shows \( (f \text{ contour-integrable-on linepath a b}) \longleftrightarrow \)
    \( ((\lambda x. f \text{ (of-real } x + \text{Im a} * i)) \text{ integrable-on } \{ \text{Re a..Re b} \}) \)
     \text{ using has-contour-integral-linepath-same-Im-iff [OF assms, of f]} \)
  by (auto simp: contour-integrable-on-def integrable-on-def)

lemma contour-integral-linepath-same-Im:
  fixes \( a \text{ b :: complex and } f :: \text{complex} \Rightarrow \text{complex} \)
  assumes \( \text{Im a = Im b Re a < Re b} \)
  shows \( \text{contour-integral (linepath a b) } f = \text{integral } \{ \text{Re a..Re b} \} (\lambda x. f \text{ (x + Im a * i)}) \)

proof (cases \( f \text{ contour-integrable-on linepath a b} \))
  case True
  thus \( \text{?thesis using has-contour-integral-linepath-same-Im-iff [OF assms, of f]} \)
    using has-contour-integral-integral has-contour-integral-unique by blast
  next
case False
thus thesis using contour-integrable-linepath-same-Im-iff [OF assms, of f] by (simp add: not-integrable-contour-integral not-integrable-integral)

qed


lemma continuous-on-compact-bound:
assumes compact A continuous-on A f
obtains B where B ≥ 0 ∧ x ∈ A ⇒ norm (f x) ≤ B
proof -
from assms (2,1) have compact (f · A) by (rule compact-continuous-image)
then obtain B where ∀x ∈ A. norm (f x) ≤ B by (auto dest!: compact-imp-bounded simp: bounded-iff)
hence max B 0 ≥ 0 and ∀x ∈ A. norm (f x) ≤ max B 0 by auto
thus thesis using that by blast

qed

interpretation cis: periodic-fun-simple cis 2 * pi by standard (simp-all add: complex-eq-iff)

lemma open-contains-cbox:
fixes x :: 'a :: euclidean-space
assumes open A x ∈ A
obtains a b where cbox a b ⊆ A x ∈ box a b ∀i ∈ Basis. a · i < b · i
proof -
from assms obtain R where R > 0 ball x R ⊆ A by (auto simp: open-contains-ball)
define r :: real where r = R / (2 * sqrt DIM('a))
from (R > 0) have [simp]: r > 0 by (auto simp: r-def)
define d :: 'a where d = r * _Topology-Euclidean-Space.One
have cbox (x − d) (x + d) ⊆ A
proof safe
fix y assume y: y ∈ cbox (x − d) (x + d)
have dist x y = sqrt (∑i∈Basis. (dist (x · i) (y · i))^2)
  by (subst euclidean-dist-l2) (auto simp: L2-set-def)
also have sqrt (∑i∈Basis. (dist (x · i) (y · i))^2) ≤ sqrt (∑i∈Basis::'a set. r^2)
  by (intro real-sqrt-le-mono sum-mono power-mono)
  (auto simp: dist-norm d-def cbox-def algebra_simps)
also have ... = sqrt (DIM('a) * r^2) by simp
also have DIM('a) * r^2 = (R / 2)^2
  by (simp add: r-def power-divide)
also have sqrt ... = R / 2
  using (R > 0) by simp
also from (R > 0) have ... < R by simp
finally have y ∈ ball x R by simp
with R show y ∈ A by blast
thus ?thesis
  using that[of x − d x + d] by (auto simp: algebra-simps d_def box_def)
qed

lemma open-contains-box:
  fixes x :: ′a :: euclidean-space
  assumes open A x ∈ A
  obtains a b where box a b ⊆ A x ∈ box a b ∀ i ∈ Basis. a · i < b · i
proof −
  from open-contains-cbox[OF assms] guess a b .
  with that[of a b] box-subset-cbox[of a b] show ?thesis by auto
qed

lemma analytic-onE-box:
  assumes f analytic-on A s ∈ A
  obtains a b where Re a < Re b Im a < Im b s ∈ box a b f analytic-on box a b
proof −
  from assms obtain r where r > 0 f holomorphic-on ball s r by (auto simp: analytic-on-def)
  with open-contains-box[of ball s r s] obtain a b
    where box a b ⊆ ball s r s ∈ box a b ∀ i ∈ Basis. a · i < b · i by auto
  moreover from r have f analytic-on ball s r by (simp add: analytic-on-open)
  ultimately show ?thesis using that[of a b] analytic-on-subset[of - ball s r box a b]
    by (auto simp: Basis-complex-def)
qed

lemma inner-image-box:
  assumes (i :: ′a :: euclidean-space) ∈ Basis
  assumes ∀ i ∈ Basis. a · i < b · i
  shows (λx. x · i) ' box a b = {a · i <..< b · i}
proof safe
  fix x assume x: x ∈ {a · i <..< b · i}
  let ?y = (∑ j ∈ Basis. (if i = j then x else (a + b) · j / 2) *R j)
  from x assms have ?y · i ∈ (λx. x · i) ' box a b
    by (intro imageI) (auto simp: box-def algebra-simps)
  also have ?y · i = (∑ j ∈ Basis. (if i = j then x else (a + b) · j / 2) * (j · i))
    by (simp add: inner-sum-left)
  also have ... = (∑ j ∈ Basis. if i = j then x else 0)
    by (intro sum.cong) (auto simp: inner-not-same-Basis assms)
  also have ... = x using assms by simp
  finally show x ∈ (λx. x · i) ' box a b .
qed (insert assms, auto simp: box-def)

lemma Re-image-box:
  assumes Re a < Re b Im a < Im b
  shows Re ' box a b = {Re a <..< Re b}
  using inner-image-box[of 1::complex a b] assms by (auto simp: Basis-complex-def)
lemma `Im-image-box`:
  assumes `Re a < Re b` \ `Im a < Im b`
  shows `Im `box a b = {Im a..<Im b}`
  using `inner-image-box[of `i::complex a b]` assms by (auto simp: `Basis-complex-def`)

lemma `inner-image-box`:
  assumes `(i :: `a :: euclidean-space) \in Basis`
  assumes `\forall i \in Basis. a \cdot i \leq b \cdot i`
  shows `(\lambda x. x \cdot i) `cbox a b = {a \cdot i..b \cdot i}`
proof safe
  fix x assume x: `x \in {a \cdot i..b \cdot i}`
  let ?y = `(\sum j \in Basis. (if i = j then x else a \cdot j) \ast_R j)`
  from x assms have ?y \cdot i \in `(\lambda x. x \cdot i) `cbox a b`
    by (intro imageI) (auto simp: `cbox-def`)
  also have ?y \cdot i = `(\sum j \in Basis. if i = j then x else a \cdot j) \ast (j \cdot i))`
    by (simp add: `inner-sum-left`)
  also have \ldots = `x` using assms by simp
  finally show `x \in (\lambda x. x \cdot i) `cbox a b` .
qed (insert assms, auto simp: `cbox-def`)

lemma `Re-image-cbox`:
  assumes `Re a \leq Re b` \ `Im a \leq Im b`
  shows `Re `cbox a b = {Re a..Re b}`
  using `inner-image-cbox[of `i::complex a b]` assms by (auto simp: `Basis-complex-def`)

lemma `Im-image-cbox`:
  assumes `Re a \leq Re b` \ `Im a \leq Im b`
  shows `Im `cbox a b = {Im a..Im b}`
  using `inner-image-cbox[of `i::complex a b]` assms by (auto simp: `Basis-complex-def`)

lemma `analytic-onE-cball`:
  assumes `f analytic-on A` \ `s \in A` \ `ub > (0::real)`
  obtains `R` where `R > 0` \ `R < ub` \ `f analytic-on cball s R`
proof –
  from assms obtain r where `r > 0` \ `f holomorphic-on ball s r`
    by (auto simp: analytic-on-def)
  hence `f analytic-on ball s r` by (simp add: analytic-on-open)
  hence `f analytic-on cball s (min (ub / 2) (r / 2))`
    by (rule analytic-on-subset, subst cball-subset-ball-iff) (use `r > 0` in auto)
  moreover have `min (ub / 2) (r / 2) > 0` and `min (ub / 2) (r / 2) < ub`
    using `r > 0` and `ub > 0` by (auto simp: min-def)
  ultimately show `?thesis` using that[of min (ub / 2) (r / 2)]
    by blast
qed
corollary analytic-pre-zeta': [analytic-intros]:
assumes f analytic-on A a > 0
shows (λx. pre-zeta a (f x)) analytic-on A
using analytic-on-compose-gen[OF assms(1) analytic-pre-zeta[of a UNIV]] assms(2)
by (auto simp: o-def)

corollary analytic-hurwitz-zeta' [analytic-intros]:
assumes f analytic-on A (⋀x. x ∈ A ⇒ f x ≠ 1) a > 0
shows (λx. hurwitz-zeta a (f x)) analytic-on A
using analytic-on-compose-gen[OF assms(1) analytic-hurwitz-zeta[of a −{1}]] assms(2,3)
by (auto simp: o-def)

corollary analytic-zeta' [analytic-intros]:
assumes f analytic-on A (⋀x. x ∈ A ⇒ f x ≠ 1)
shows (λx. zeta (f x)) analytic-on A
using analytic-on-compose-gen[OF assms(1) analytic-zeta[of −{1}]] assms(2)
by (auto simp: o-def)

lemma logderiv-zeta-analytic: (λs. deriv zeta s / zeta s)
analytic-on {s. Re s ≥ 1} − {1}
using zeta-Re-ge-1-nonzero by (auto intro!: analytic-intros)

lemma cis-pi-half [simp]: cis (pi / 2) = i
by (simp add: complex-eq-iff)

lemma mult-real-sqrt: x ≥ 0 ⇒ x * sqrt y = sqrt (x * 2 * y)
by (simp add: real-sqrt-mult)

lemma arcsin-pos: x ∈ {0<..<1} ⇒ arcsin x > 0
using arcsin-less-arcsin[of 0 x] by simp

lemmas analytic-imp-holomorphic' = holomorphic-on-subset[OF analytic-imp-holomorphic]

lemma residue-simple':
assumes open s 0 ∈ s f holomorphic-on s
shows residue (λw. f w / w) 0 = f 0
using residue-simple[of s 0 f] assms by simp

lemma fds-converges-cong:
assumes eventually (λn. fds-nth f n = fds-nth g n) at-top s = s'
shows fds-converges f s ←→ fds-converges g s'
unfolding fds-converges-def
by (intro summable-cong eventually-mono[OF assms(1)]) (simp-all add: assms)

lemma fds-abs-converges-cong:
assumes eventually (λn. fds-nth f n = fds-nth g n) at-top s = s'

14
shows \( \text{fds-abs-converges } f \leftrightarrow \text{fds-abs-converges } g \) \\
unfolding \( \text{fds-abs-converges-def} \) \\
by (intro summable-cong eventually-mono[of assms]) (simp-all add: assms)

lemma conv-abscissa-cong:
assumes eventually \((\lambda n. \text{fds-nth } f \ n = \text{fds-nth } g \ n)\) at-top
shows \( \text{conv-abscissa } f = \text{conv-abscissa } g \)
proof —
  have \( \text{fds-converges } f = \text{fds-converges } g \)
    by (intro ext fds-converges-cong assms refl)
  thus \?thesis by (simp add: conv-abscissa-def)
qed

lemma abs-conv-abscissa-cong:
assumes eventually \((\lambda n. \text{fds-nth } f \ n = \text{fds-nth } g \ n)\) at-top
shows \( \text{abs-conv-abscissa } f = \text{abs-conv-abscissa } g \)
proof —
  have \( \text{fds-abs-converges } f = \text{fds-abs-converges } g \)
    by (intro ext fds-abs-converges-cong assms refl)
  thus \?thesis by (simp add: abs-conv-abscissa-def)
qed

definition fds-remainder where
\( \text{fds-remainder } m = \text{fds-subseries } (\lambda n. n > m) \)

lemma fds-nth-remainder: \( \text{fds-nth } (\text{fds-remainder } m \ f) = (\lambda n. \text{if } n > m \text{ then } \text{fds-nth } f \ n \text{ else } 0) \)
  by (simp add: fds-remainder-def fds-subseries-def fds-nth-fds')

lemma fds-converges-remainder-iff [simp]:
\( \text{fds-converges } (\text{fds-remainder } m \ f) \ s \leftrightarrow \text{fds-converges } f \ s \)
  by (intro fds-converges-cong eventually-mono[of eventually-gt-at-top[of m]])
    (auto simp: fds-nth-remainder)

lemma fds-abs-converges-remainder-iff [simp]:
\( \text{fds-abs-converges } (\text{fds-remainder } m \ f) \ s \leftrightarrow \text{fds-abs-converges } f \ s \)
  by (intro fds-abs-converges-cong eventually-mono[of eventually-gt-at-top[of m]])
    (auto simp: fds-nth-remainder)

lemma fds-converges-remainder [intro]:
\( \text{fds-converges } f \ s \Longrightarrow \text{fds-converges } (\text{fds-remainder } m \ f) \ s \)
and fds-abs-converges-remainder [intro]:
\( \text{fds-abs-converges } f \ s \Longrightarrow \text{fds-abs-converges } (\text{fds-remainder } m \ f) \ s \)
by simp-all

lemma conv-abscissa-remainder [simp]:
\( \text{conv-abscissa } (\text{fds-remainder } m \ f) = \text{conv-abscissa } f \)
by (intro conv-abscissa-cong eventually-mono[of eventually-gt-at-top[of m]])

15
lemma abs-conv-abscissa-remainder [simp]:
  abs-conv-abscissa (fds-remainder m f) = abs-conv-abscissa f
by (intro abs-conv-abscissa-cong eventually-mono[OF eventually-gt-at-top[of m]])
(auto simp: fds-remainder-def)

lemma eval-fds-remainder:
eval-fds (fds-remainder m f) s = (∑ n. fds-nth f (n + Suc m) / nat-power (n + Suc m) s)
(is - = suminf (λn. ?f (n + Suc m)))
proof (cases fds-converges f s)
case False
hence ¬fds-converges (fds-remainder m f) s by simp
hence (λx. (λn. fds-nth (fds-remainder m f) n / nat-power n s) sums x) = (λ-. False)
  by (auto simp: fds-converges-def summable-def)
  hence eval-fds (fds-remainder m f) s = (THE -. False)
  by (simp add: eval-fds-def suminf-def)
moreover have ¬summable (λn. ?f (n + Suc m)) unfolding fds-converges-def
  by (subst summable-iff-shift) auto
hence (λx. (λn. ?f (n + Suc m)) sums x) = (λ-. False)
  by (auto simp: summable-def)
  hence suminf (λn. ?f (n + Suc m)) = (THE -. False)
  by (simp add: suminf-def)
ultimately have ?thesis by simp

next
case True
hence *: fds-converges (fds-remainder m f) s by simp
have eval-fds (fds-remainder m f) s = (∑ n. fds-nth (fds-remainder m f) n / nat-power n s)
  unfolding eval-fds-def ..
also have .. = (∑ n. fds-nth (fds-remainder m f) (n + Suc m) / nat-power (n + Suc m) s)
  using * unfolding fds-converges-def
  by (subst suminf-minus-initial-segment) (auto simp: fds-nth-remainder)
also have (λn. fds-nth (fds-remainder m f) (n + Suc m)) = (λn. fds-nth f (n + Suc m))
  by (intro ext) (auto simp: fds-nth-remainder)
finally have ?thesis .
qed

lemma fds-truncate-plus-remainder: fds-truncate m f + fds-remainder m f = f
by (intro fds-eqI) (auto simp: fds-truncate-def fds-remainder-def fds-subseries-def)

lemma holomorphic-fds-eval' [holomorphic-intros]:
  assumes g holomorphic-on A ∧ x ∈ A → Re (g x) > conv-abscissa f
\( \lambda x. \text{eval-fds } f \ (g \ x) \) holomorphic-on \( A \)

using holomorphic-on-compose-gen[\( \text{OF assms}(1) \) holomorphic-fds-eval[\( \text{OF order.refl, of } f ]\) assms(2)

by (auto simp: o-def)

lemma analytic-fds-eval' [analytic-intros]:
assumes \( g \) analytic-on \( A \) \( \forall x. x \in A \implies \Re (g \ x) > \text{conv-abscissa } f \)

shows \( \lambda x. \text{eval-fds } f \ (g \ x) \) analytic-on \( A \)

using analytic-on-compose-gen[\( \text{OF assms}(1) \) analytic-fds-eval[\( \text{OF order.refl, of } f ]\) assms(2)

by (auto simp: o-def)

lemma homotopic-loopsI:
fixes \( h \) :: \( \text{real} \times \text{real} \Rightarrow - \)

assumes continuous-on \( \{0..1\} \times \{0..1\} \) \( h \)
\( h \ (\{0..1\} \times \{0..1\}) \subseteq s \)
\( \forall x. x \in \{0..1\} \implies h \ (0, x) = p \ x \)
\( \forall x. x \in \{0..1\} \implies h \ (1, x) = q \ x \)
\( \forall x. x \in \{0..1\} \implies \text{pathfinish } (h \circ \text{Pair } x) = \text{pathstart } (h \circ \text{Pair } x) \)

shows \( \text{homotopic-loops } s \ p \ q \)

using assms unfolding homotopic-loops by (intro exI [of - h]) auto

lemma continuous-on-linepath [continuous-intros]:
assumes continuous-on \( A \) \( a \) continuous-on \( A \) \( b \) continuous-on \( A \) \( f \)

shows continuous-on \( A \) \( \lambda x. \text{linepath } (a \ x) \ (b \ x) \ (f \ x) \)

using assms by (auto simp: linepath-def intro: continuous-intros assms)

lemma continuous-on-part-circlepath [continuous-intros]:
assumes continuous-on \( A \) \( c \) continuous-on \( A \) \( r \) continuous-on \( A \) \( a \) continuous-on \( A \) \( b \)
continuous-on \( A \) \( f \)

shows continuous-on \( A \) \( \lambda x. \text{part-circlepath } (c \ x) \ (r \ x) \ (a \ x) \ (b \ x) \ (f \ x) \)

using assms by (auto simp: part-circlepath-def intro!: continuous-intros assms)

lemma homotopic-loops-part-circlepath:
assumes sphere \( c \) \( r \subseteq A \) and \( r \geq 0 \) and
\( b1 = a1 + 2 \ast \text{of-int } k \ast \pi \) and \( b2 = a2 + 2 \ast \text{of-int } k \ast \pi \)

shows homotopic-loops \( A \) \( (\text{part-circlepath } c \ r \ a1 \ b1) \ (\text{part-circlepath } c \ r \ a2 \ b2) \)

proof
  define \( h \) where \( h = (\lambda (x,y). \text{part-circlepath } c \ r \ (\text{linepath } a1 \ a2 \ x) \ (\text{linepath } b1 \ b2 \ x) \ y) \)
  show \( \text{thesis} \)
  proof (rule homotopic-loopsI)
    show continuous-on \( \{0..1\} \times \{0..1\} \) \( h \)
    by (auto simp: h-def case-prod-unfold intro!: continuous-intros)
  next
    from assms have \( h \ (\{0..1\} \times \{0..1\}) \subseteq \text{sphere } c \ r \)
    by (auto simp: h-def part-circlepath-def dist-norm norm-mult)
    also have \( \ldots \subseteq A \) by fact
finally show $h' (\{0..1\} \times \{0..1\}) \subseteq A$.

next

fix $x :: \text{real}$ assume $x: x \in \{0..1\}$
show $h (0, x) =$ part-circlepath $c r a1 b1 x$ and $h (1, x) =$ part-circlepath $c r a2 b2 x$

by (simp-all add: h-def linepath-def)

have $\text{cis } (\pi * (\text{real-of-int } k * 2)) = 1$
using cis.

thus $\text{pathfinish } (h \circ \text{Pair } x) = \text{pathstart } (h \circ \text{Pair } x)$

by (simp add: h-def o-def exp-eq-polar linepath-def algebra-simps cis-mult [symmetric] cis-divide [symmetric] assum)

qed

qed

lemma homotopic-pathsI:
fixes $h :: \text{real} \times \text{real} \Rightarrow -$  
assumes $\text{continuous-on } (\{0..1\} \times \{0..1\}) h$
assumes $h' (\{0..1\} \times \{0..1\}) \subseteq s$
assumes $\forall x. x \in \{0..1\} \implies h (0, x) = p x$
assumes $\forall x. x \in \{0..1\} \implies h (1, x) = q x$
assumes $\forall x. x \in \{0..1\} \implies \text{pathstart } (h \circ \text{Pair } x) = \text{pathstart } p$
assumes $\forall x. x \in \{0..1\} \implies \text{pathfinish } (h \circ \text{Pair } x) = \text{pathfinish } p$

shows $\text{homotopic-paths } s p q$
using assum unfolding homotopic-paths by (intro exI [of - h]) auto

lemma part-circlepath-conv-subpath:
part-circlepath $c r a b = \text{subpath } (a / (2 * \pi)) (b / (2 * \pi)) (\text{circlepath } c r)$

by (simp add: part-circlepath-def circlepath-def subpath-def linepath-def algebra-simps exp-eq-polar)

lemma homotopic-paths-part-circlepath:
assumes $a \leq b \leq c$
assumes $\text{path-image } (\text{part-circlepath } C r a c) \subseteq A \ r \geq 0$
shows $\text{homotopic-paths } A (\text{part-circlepath } C r a c)$

(is homotopic-paths - ?g ($\text{thesis}$

proof (cases $a = c$)
case False
with assum have $a \leq c$ by simp

define slope where slope $= (b - a) / (c - a)$

from assum and ($a < c$) have slope: slope $\in \{0..1\}$

by (auto simp: field-simps slope-def)

define $f :: \text{real} \Rightarrow \text{real}$ where $f = \text{linepath } 0 \text{ slope } + + + \text{ linepath } \text{slope } 1$

show $\text{thesis}$

proof (rule homotopic-paths-reparametrize)
fix $t :: \text{real}$ assume $t: t \in \{0..1\}$

show ($\text{thesis}$

18
proof (cases \( t \leq 1 / 2 \))

\begin{itemize}
\item case True
  \begin{itemize}
  \item hence \( ?g (f t) = C + r * \text{cis} \((1 - f t) * a + f t * c) \)
  \item by \((\text{simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def})\)
  \item also from True \((a < c)\) have \((1 - f t) * a + f t * c = (1 - 2 * t) * a + 2 * t * b\)
  \item unfolding f-def slope-def linepath-def joinpaths-def
  \item by \((\text{simp add: divide-simps del: div-mult-self3 div-mult-self4 div-mult-self2 div-mult-self1})\)
  \item also from True have \(C + r * \text{cis} \((1 - f t) * a + f t * c) = (?h1 +++ ?h2) t\)
  \item finally show \(?thesis ..\)
\end{itemize}
\end{itemize}

next

\begin{itemize}
\item case False
  \begin{itemize}
  \item hence \( ?g (f t) = C + r * \text{cis} \((1 - f t) * a + f t * c) \)
  \item by \((\text{simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def})\)
  \item also from False \((a < c)\) have \((1 - f t) * a + f t * c = (2 - 2 * t) * b + (2 * t - 1) * c\)
  \item unfolding f-def slope-def linepath-def joinpaths-def
  \item by \((\text{simp add: divide-simps del: div-mult-self3 div-mult-self4 div-mult-self2 div-mult-self1})\)
  \item also from False have \(C + r * \text{cis} \((1 - f t) * a + f t * c) = (?h1 +++ ?h2) t\)
  \item finally show \(?thesis ..\)
\end{itemize}
\end{itemize}

qed

next

\begin{itemize}
\item from slope have path-image \( f \subseteq \{0..1\} \)
  \item by \((\text{auto simp: f-def path-image-join closed-segment-eq-real-ivl})\)
  \item thus \( f ' \{0..1\} \subseteq \{0..1\} \) by \((\text{simp add: path-image-def})\)
\end{itemize}

next

\begin{itemize}
\item have path \( f \) unfolding f-def by auto
  \item thus continuous-on \( \{0..1\} f \) by \((\text{simp add: path-def})\)
\end{itemize}

qed (insert assms, auto simp: f-def joinpaths-def linepath-def)

next

\begin{itemize}
\item case \([simp]: True\)
  \item with assms have \([simp]: b = c \) by auto
\end{itemize}

lemma has-contour-integral-mirror-iff:
\begin{itemize}
\item assumes \(\text{valid-path } g\)
\item shows \((f \text{has-contour-integral } I) (-g) \iff ((\lambda x. -f (-x)) \text{has-contour-integral } I) g\)
\end{itemize}

proof –
from assms have \( g \) piecewise-differentiable-on \{0..1\}
by (auto simp: valid-path-def piecewise-C1-imp-differentiable)
then obtain \( S \) where \( S: \text{finite} \ S \land x \in \{0..1\} \Rightarrow g \text{ differentiable at } x \)
within \{0..1\}
  unfolding piecewise-differentiable-on-def by blast
have \( S': g \text{ differentiable at } x \) if \( x \in \{0..1\} - ((\{0, 1\} \cup S) \text{ for } x) \)
proof
  from that have \( x \in \text{interior} \{0..1\} \) by auto
with \( S(2)[of \ x] \) that show \(?thesis\) by (auto simp: at-within-interior[of - \{0..1\}])
qed

have \( (f \text{ has-contour-integral } I) (-g) \leftrightarrow ((\lambda x. f (- g x) \ast \text{vector-derivative} (-g) \ (at \ x)) \text{ has-integral } I) \{0..1\} \)
by (simp add: has-contour-integral)
also have \(. . . \leftrightarrow ((\lambda x. -f (- g x) \ast \text{vector-derivative} g \ (at \ x)) \text{ has-integral } I) \{0..1\} \)
by (intro has-integral-spike-finite-eq[of \( S \cup \{0, 1\} \)]
  (insert \( \text{finite} \ S \); \( S' \), auto simp: o-def fun-Compl-def)
also have \(. . . \leftrightarrow ((\lambda x. -f (- x)) \text{ has-contour-integral } I) \ g \)
by (simp add: has-contour-integral)
finally show \(?thesis\).
qed

lemma contour-integral-on-mirror-iff:
assumes valid-path \( g \)
shows \( f \text{ contour-integrable-on } (-g) \leftrightarrow \ (\lambda x. -f (- x)) \text{ contour-integrable-on } g \)
by (auto simp: contour-integrable-on-def has-contour-integral-mirror-iff assms)

lemma contour-integral-mirror:
assumes valid-path \( g \)
shows \( \text{contour-integral } (-g) \ f = \text{contour-integral } (\lambda x. -f (- x)) \ g \)
proof (cases \( f \text{ contour-integrable-on } (-g) \))
case True
then obtain \( I \) where \( I: (f \text{ has-contour-integral } I) (-g) \)
by (auto simp: contour-integrable-on-def)
also note has-contour-integral-mirror-iff[OF assms]
finally have \((\lambda x. -f (- x)) \text{ has-contour-integral } I) \ g \).
with \( I \) show \(?thesis\) using contour-integral-unique by blast
next
case False
hence \(-((\lambda x. -f (- x)) \text{ contour-integrable-on } g \)
by (auto simp: contour-integrable-on-mirror-iff assms)
from False and this show \(?thesis\)
by (simp add: not-integrable-contour-integral)
qed

lemma contour-integrable-neg-iff:
\( (\lambda x. -f x) \text{ contour-integrable-on } g \leftrightarrow f \text{ contour-integrable-on } g \)
using \textit{contour-integrable-neg[of f g]} \textit{contour-integrable-neg[of \lambda x. -f x g]} by auto

lemma \textit{contour-integral-neg}:
shows contour-integral \( g (\lambda x. -f x) \) = \(-\text{contour-integral} \ g \ f \)
proof (cases \( f \) \textit{contour-integrable-on} \( g \))
case True
thus \( \text{thesis} \) by (simp add: \textit{contour-integral-neg})
next
case False
hence \(~(\lambda x. -f x) \textit{contour-integrable-on} \ g \) by (simp add: \textit{contour-integrable-neg-iff})
with False show \( \text{thesis} \) by (simp add: \textit{not-integrable-contour-integral})
qed

lemma \textit{minus-cis}:
\(-\text{cis} \ x = \text{cis} \ (x + \pi)\)
by (simp add: \textit{complex-eq-iff})

lemma \textit{path-image-part-circlepath-subset}:
assumes \( a \leq a' \) \( a' \leq b' \) \( b' \leq b \)
shows \( \text{path-image} \ (\text{part-circlepath} \ c \ r \ a \ b) \subseteq \text{path-image} \ (\text{part-circlepath} \ c \ r \ a' \ b') \)
using \textit{assms} by (subst (1 2) \textit{path-image-part-circlepath}) auto

lemma \textit{part-circlepath-mirror}:
assumes \( a' = a + \pi + 2 \ast \pi \ast \text{of-int} \ k \) \( b' = b + \pi + 2 \ast \pi \ast \text{of-int} \ k \)
shows \( \text{part-circlepath} \ c \ r \ a \ b = \text{part-circlepath} \ c' \ r \ a' \ b' \)
proof
fix \( x :: \text{real} \)
have \( \text{part-circlepath} \ c' \ r \ a' \ b' \ x = c' + r \ast \text{cis} \ (\text{linepath} \ a \ b \ x + \pi + k \ast (2 \ast \pi)) \)
by (simp add: \textit{part-circlepath-def} \textit{exp-eq-polar} \textit{assms} \textit{linepath-translate-right} mult-ac)
also have \( \text{cis} \ (\text{linepath} \ a \ b \ x + \pi + k \ast (2 \ast \pi)) = \text{cis} \ (\text{linepath} \ a \ b \ x + \pi) \)
by (rule \textit{cis.plus-of-int})
also have \( \ldots = -\text{cis} \ (\text{linepath} \ a \ b \ x) \)
by (simp add: \textit{minus-cis})
also have \( c' + r \ast \ldots = -\text{part-circlepath} \ c \ r \ a \ b \ x \)
by (simp add: \textit{part-circlepath-def} \textit{assms} \textit{exp-eq-polar})
finally show \( \text{part-circlepath} \ c \ r \ a \ b \ x = \text{part-circlepath} \ c' \ r \ a' \ b' \ x \)
by simp
qed

lemma \textit{path-mirror} \[\text{intro}]: \( \text{path} \ (g :: - \Rightarrow 'b::topological-group-add) \Longrightarrow \text{path} \ (-g) \)
by (auto simp: \textit{path-def} \textit{intro}: \textit{continuous-intros})

lemma \textit{path-mirror-iff} \[\text{simp}]: \( \text{path} \ (-g :: - \Rightarrow 'b::topological-group-add) \iff \text{path} \ g \)

21
lemma valid-path-mirror [intro]: valid-path g \implies valid-path (−g) by (auto simp: valid-path-def fun-Compl-def)

lemma valid-path-mirror-iff [simp]: valid-path (−g) \iff valid-path g using valid-path-mirror[of g] valid-path-mirror[of −g] by (auto simp: fun-Compl-def)

lemma pathstart-mirror [simp]: pathstart (−g) = −pathstart g and pathfinish-mirror [simp]: pathfinish (−g) = −pathfinish g by (simp-all add: pathstart-def pathfinish-def)

lemma path-image-mirror: path-image (−g) = uminus ' path-image g by (auto simp: path-image-def)

lemma contour-integral-bound-part-circlepath: assumes f contour-integrable-on part-circlepath c r a b assumes B ≥ 0 r ≥ 0 \( \forall x. x \in \text{path-image} (\text{part-circlepath} c r a b) \implies \text{norm} (f x) \leq B \) shows \( \text{norm} (\text{contour-integral} (\text{part-circlepath} c r a b) f) \leq B * r * |b - a| \) proof - let ?I = integral \{0..1\} (\( \lambda x. f (\text{part-circlepath} c r a b x) * i * \text{of-real} (r * (b - a)) \) * exp (i * linopath a b x)) have \( \text{norm} ?I \leq \text{integral} \{0..1\} (\( \lambda x::\text{real}. B * 1 * (r * |b - a|) * 1 \)) \) proof (rule integral-norm-bound-integral, goal-cases) case 1 with assms(1) show ?case by (simp add: contour-integrable-on vector-derivative-part-circlepath mult-ac) next case (3 x) with assms(2 −) show ?case unfolding norm-mult norm-of-real abs-mult by (intro mult-mono) (auto simp: path-image-def) qed auto also have ?I = contour-integral (\text{part-circlepath} c r a b) f by (simp add: contour-integral-integral vector-derivative-part-circlepath mult-ac) finally show ?thesis by simp qed

lemma contour-integral-spike-finite-simple-path: assumes finite A simple-path g \( g = g' \bigwedge x. x \in \text{path-image} g - A \implies f x = f' x \) shows contour-integral g f = contour-integral g' f' unfolding contour-integral-integral proof (rule integral-spike) have finite (\( g = A \cap \{0<..<1\} \)) using (simple-path g) (finite A) by (intro finite-vimage-IntI simple-path-inj-on) auto hence finite \( \{0, 1\} \cup g = -A \cap \{0<..<1\} \) by auto thus negligible \( \{0, 1\} \cup g = -A \cap \{0<..<1\} \) by (rule negligible-finite) next
fix $x$ assume $x \in \{0..1\} - (\{0, 1\} \cup g^{-1} A \cap \{0<..<1\})$
hence $g x \in \text{path-image} g - A$ by (auto simp: path-image-def)
from assms(4)[OF this] and assms(3)
  show $f' (g' x) \ast \text{vector-derivative} g' (at x) = f (g x) \ast \text{vector-derivative} g (at x)$ by simp
qed

proposition contour-integral-bound-part-circlepath-strong:
assumes $f$: $f$ contour-integrable-on part-circlepath $z r s t$
and finite $k$
and $B$: $0 \leq B \leq r s \leq t$
shows $\text{cmod} (\text{contour-integral} (\text{part-circlepath} z r s t) f) \leq B \ast r \ast (t - s)$
proof –
from $f$ have $(f$ has-contour-integral contour-integral (part-circlepath $z r s t) f)$
  (part-circlepath $z r s t)$
by (rule has-contour-integral-integral)
from has-contour-integral-bound-part-circlepath-strong[OF this assms(2-)] show $\text{thesis}$ by auto
qed

lemma cos-le-zero:
assumes $x \in \{\pi/2..3\pi/2\}$
shows $\cos x \leq 0$
proof –
  have $\cos x = -\cos (x - \pi)$ by (simp add: cos-diff)
moreover from assms have $\cos (x - \pi) \geq 0$
  by (intro cos-ge-zero) auto
ultimately show $\text{thesis}$ by simp
qed

lemma cos-le-zero': $x \in \{-3\pi/2..-\pi/2\} \implies \cos x \leq 0$
using cos-le-zero[of $-x$] by simp

lemma cis-minus-pi-half [simp]: $\text{cis} (- (\pi / 2)) = -i$
  by (simp add: complex-eq-iff)

lemma winding-number-join-pos-combined':
  [valid-path $\gamma 1 \land z \notin \text{path-image} \gamma 1 \land 0 < \text{Re} (\text{winding-number} \gamma 1 z);$
  valid-path $\gamma 2 \land z \notin \text{path-image} \gamma 2 \land 0 < \text{Re} (\text{winding-number} \gamma 2 z);$
  pathfinish $\gamma 1 = \text{pathstart} \gamma 2]$
  $\implies$ valid-path($\gamma 1 +++ \gamma 2 \land z \notin \text{path-image}(\gamma 1 +++ \gamma 2) \land 0 < \text{Re}(\text{winding-number}(\gamma 1 +++ \gamma 2) z)$
by (simp add: valid-path-join path-image-join winding-number-join valid-path-imp-path)

lemma Union-atLeastAtMost-real-of-nat:
  assumes $a < b$
sows $\{\bigcup n \in \{a..b\}, \{\text{real} n..\text{real} (n + 1)\}\} = \{\text{real} a..\text{real} b\}$
proof (intro equalityI subsetI)
fix $x$ assume $x$: $x \in \{\text{real} a..\text{real} b\}$

23
thus \( x \in (\bigcup_{n \in \{a \cdot< b\}} \{\text{real } n \cdot \text{real } (n + 1)\}) \)

proof (cases \( x = \text{real } b \))
case True
  with assms show \(?thesis\) by (auto intro!: bexI[of \(- b - 1\)])
next
case False
with \( x \) have \( x \geq \text{real } a \cdot< \text{real } b \) by simp-all
hence \( x \geq (\text{real } \lfloor x \rfloor) \) \( \leq \text{real } (\text{Suc } (\text{nat } \lfloor x \rfloor)) \) by linarith+
moreover from \( x \) have \( \text{nat } \lfloor x \rfloor \geq a \cdot \text{nat } \lfloor x \rfloor < b \) by linarith+
ultimately have \( \exists n \in \{a \cdot< b\}. \ x \in \{\text{real } n \cdot \text{real } (n + 1)\} \)
  by (intro bexI[of \(- n \cdot \text{nat } \lfloor x \rfloor\)] simp-all
thus \(?thesis\) by blast
qed
qed auto

lemma nat-sum-has-integral-floor:
  fixes \( f \) :: \( \text{nat} \Rightarrow 'a \) :: banach
  assumes \( \text{mn} \cdot m < n \)
  shows \((\lambda x. \ f \ (\text{nat } \lfloor x \rfloor)) \text{-has-integral } \text{sum } f \ \{m \cdot< n\}\ \{\text{real } m \cdot \text{real } n\}\)
proof –
define \( D \) where \( D = (\lambda i. \ \{\text{real } i \cdot \text{real } (\text{Suc } i)\}) \cdot \{m \cdot< n\}\)
  have \( D : D \text{-division-of } \{m \cdot n\}\)
    using Union-atLeastAtMost-real-of-nat[OF \( mn \)] by (simp add: division-of-def D-def)
  have \((\lambda x. \ f \ (\text{nat } \lfloor x \rfloor)) \text{-has-integral } (\sum X \in D. \ f \ (\text{nat } [\text{Inf } X] ))\ \{\text{real } m \cdot \text{real } n\}\)
proof (rule has-integral-combine-division)
  fix \( X \) assume \( X : X \in D \)
  have \( \text{nat } \lfloor x \rfloor = \text{nat } [\text{Inf } X] \) if \( x \in X \) \(- \{\text{Sup } X\}\) for \( x \)
    using that \( X \) by (auto simp: D-def nat-eq-iff floor-eq-iff)
  hence \((\lambda x. \ f \ (\text{nat } \lfloor x \rfloor)) \text{-has-integral } f \ (\text{nat } [\text{Inf } X] ))\ X \leftarrow
    ((\lambda x. \ f \ (\text{nat } [\text{Inf } X] )) \text{-has-integral } f \ (\text{nat } [\text{Inf } X] ))\ X \) using \( X \)
    by (intro has-integral-spike-eq[of \{\text{Sup } X\}] auto
    also from \( X \) have \ldots using has-integral-const-real[of \( f \ (\text{nat } [\text{Inf } X] )) \text{Inf } X \)
    Sup \( X \)
    by (auto simp: D-def)
  finally show \((\lambda x. \ f \ (\text{nat } \lfloor x \rfloor)) \text{-has-integral } f \ (\text{nat } [\text{Inf } X] ))\ X \ .
  qed
  fact+
  also have \((\sum X \in D. \ f \ (\text{nat } [\text{Inf } X] ))\) = \((\sum k \in \{m \cdot< n\}. \ f k)\)
  unfolding D-def by (subst sum.reindex) (auto simp: inj-on-def nat-add-distrib)
  finally show \(?thesis\) .
qed

lemma nat-sum-has-integral-ceiling:
  fixes \( f \) :: \( \text{nat} \Rightarrow 'a \) :: banach
  assumes \( \text{mn} \cdot m < n \)
  shows \((\lambda x. \ f \ (\text{nat } \lfloor x \rfloor)) \text{-has-integral } \text{sum } f \ \{m \cdot< n\}\ \{\text{real } m \cdot \text{real } n\}\)
proof –
define \( D \) where \( D = (\lambda i. \ \{\text{real } i \cdot \text{real } (\text{Suc } i)\}) \cdot \{m \cdot< n\}\)
have D: D division-of \{m..n\}
using Union-atLeastAtMost-real-of-nat[OF mn] by (simp add: division-of-def D-def)
have ((\lambda x. f (nat \lfloor x \rfloor)) has-integral (\sum_{x \in D} f (nat \lceil Sup X \rceil))) \{ real m..real n \}
proof (rule has-integral-combine-division)
  fix X assume X: X \in D
  have nat \lfloor x \rfloor = nat \lfloor Sup X \rfloor if x \in X \setminus \{ Inf X \} for x
  using that X by (auto simp: D-def nat-eq-iff ceiling-eq-iff)
  hence ((\lambda x. f (nat \lfloor x \rfloor)) has-integral f (nat \lceil Sup X \rceil)) X \longleftrightarrow
  ((\lambda x. f (nat \lfloor Sup X \rfloor)) has-integral f (nat \lceil Sup X \rceil)) X using X
  by (intro has-integral-spike-eq[of (Inf X)]) auto
  also have \ldots using has-integral-const-real[of f (nat \lceil Sup X \rceil)] Inf X Sup X
  by (auto simp: D-def)
  finally show ((\lambda x. f (nat \lfloor x \rfloor)) has-integral f (nat \lceil Sup X \rceil)) X .
qed

lemma zeta-partial-sum-le:
fixes x :: real and m :: nat
assumes x: x \in \{0..<1\}
shows (\sum_{k=1..m} real k powr (x - 1)) \leq real m powr x / x
proof
  consider m = 0 | m = 1 | m > 1 by force
  thus \?thesis
proof cases
  assume m: m > 1
  hence \{1..m\} = insert 1 \{1..<m\} by auto
  also have (\sum_{k \in \ldots} real k powr (x - 1)) = 1 + (\sum_{k \in \{1..<m\}} real k powr (x - 1))
  by simp
  also have (\sum_{k \in \{1..<m\}} real k powr (x - 1)) \leq real m powr x / x - 1 / x
  proof (rule has-integral-le)
    show ((\lambda t. (nat \lfloor t \rfloor) powr (x - 1)) has-integral (\sum_{n \in \{1..<m\}} n powr (x - 1))) \{ real 1..m \}
    using m by (intro nat-sum-has-integral-ceiling) auto
  next
    have ((\lambda t powr (x - 1)) has-integral (real m powr x / x - real 1 powr x / x))
    \{ real 1..real m \}
    by (intro fundamental-theorem-of-calculus)
      (insert x m, auto simp flip: has-field-derivative-iff-has-vector-derivative intral!: derivative-eq-intros)
thus \((\lambda t. t \text{ powr} (x - 1)) \text{ has-integral} (\text{real m powr} x / x - 1 / x)) \{\text{real 1..real m}\}
by simp
qed (insert x, auto intro!: powr-mono2')
also have \(1 + (\text{real m powr} x / x - 1 / x) \leq \text{real m powr} x / x\)
using x by (simp add: field-simps)
finally show ?thesis by simp

qed (use assms in auto)

lemma zeta-partial-sum-le':
fixes x :: real and m :: nat
assumes x: \(x > 0\) and m: \(m > 0\)
shows \((\sum n=1..m. \text{real n powr} (x - 1)) \leq m \text{ powr} x / x\)
proof (cases x > 1)
case False
with assms have \((\sum n=1..m. \text{real n powr} (x - 1)) \leq m \text{ powr} x / x\)
by (intro zeta-partial-sum-le') auto
also have \(\ldots \leq m \text{ powr} x \ast (1 / x + 1 / m)\)
using assms by (simp add: field-simps)
finally show ?thesis .

next
case True
have \((\sum n\in\{1..m\}. \text{n powr} (x - 1)) = (\sum n\in\text{insert m} \{0..<m\}. \text{n powr} (x - 1))\)
by (intro sum.mono-neutral-left) auto
also have \(\ldots = m \text{ powr} (x - 1) + (\sum n\in\{0..<m\}. \text{n powr} (x - 1))\) by simp
also have \((\sum n\in\{0..<m\}. \text{n powr} (x - 1)) \leq \text{real m powr} x / x\)
proof (rule has-integral-le)
show \((\lambda t. (\text{nat} \lfloor t \rfloor) \text{ powr} (x - 1)) \text{ has-integral} (\sum n\in\{0..<m\}. \text{n powr} (x - 1))\) \{\text{real 0..m}\}
using m by (intro nat-sum-has-integral-floor) auto
next
show \((\lambda t. t \text{ powr} (x - 1)) \text{ has-integral} (\text{real m powr} x / x)\) \{\text{real 0..real m}\}
using has-integral-powr-from-0[of x - 1] x by auto
next
fix t assume t \in \{\text{real 0..real m}\}
with x: \(x > 1\) show \(\text{real} \ (\text{nat} \lfloor t \rfloor) \text{ powr} (x - 1) \leq t \text{ powr} (x - 1)\)
by (cases t = 0) (auto intro: powr-mono2)
qed

also have \(m \text{ powr} (x - 1) + m \text{ powr} x / x = m \text{ powr} x \ast (1 / x + 1 / m)\)
using m x by (simp add: powr-diff field-simps)
finally show ?thesis by simp

qed

lemma natfun-bigo-1E:
assumes \((f :: \text{nat} \Rightarrow \cdot) \in O(\lambda-. \ 1)\)
obtains C where \(C \geq \text{lb} \ \\land n. \ \text{norm} \ (f \ n) \leq C\)
proof –

26
from assms obtain $CN$ where $\forall n \geq N. \text{norm}(f n) \leq C$
  by (auto elim!: landau-o.bigE simp: eventually-at-top-linorder)

hence $*: \text{norm}(f n) \leq \text{Max} ([\{C, lb\} \cup (\text{norm} f \circ \{..<N\})])$ for $n$
  by (cases $n \geq N$) (subst Max-ge-iff; force simp: image-iff)+

moreover have $\text{Max} ([\{C, lb\} \cup (\text{norm} f \circ \{..<N\})]) \geq lb$
  by (intro Max.coboundedI) auto

ultimately show $\exists \text{thesis using that by blast}$
qed

lemma natfun-bigo-iff-Bseq: $f \in O(\lambda^- 1) \iff \text{Bseq } f$
proof
  assume $\text{Bseq } f$
  then obtain $C$ where $C > 0 \land n. \text{norm}(f n) \leq C$ by (auto simp: Bseq-def)
  thus $f \in O(\lambda^- 1)$ by (intro bigol[of $-C$]) auto

next
  assume $f \in O(\lambda^- 1)$
  from natfun-bigo-1E[OF this, where $lb = 1$] obtain $C$ where $C \geq 1 \land n. \text{norm}(f n) \leq C$
  by auto
  thus $\text{Bseq } f$ by (auto simp: Bseq-def intro!: exI[of $-C$])
qed

lemma enn-decreasing-sum-le-set-nn-integral:
fixes $f :: \text{real} \Rightarrow ennreal$
assumes $\text{decreasing: } \forall x y. 0 \leq x \implies x \leq y \implies y \leq f x$
shows $(\sum n. f(\text{real}(\text{Suc } n))) \leq \text{set-nn-integral } \text{lborel } \{0..\} f$
proof
  have $(\sum n. (f(\text{Suc } n))) = (\sum n. \sum x(\text{real } n..<.. \text{real}(\text{Suc } n)). (f(\text{Suc } n)) \partial \text{lborel})$
  by (subst nn-integral-cmul-indicator) auto
  also have $\text{nat } [x] = \text{Suc } n$ if $x \in \{\text{real } n..<.. \text{real}(\text{Suc } n)\}$ for $x n$
  using that by (auto simp: nat-eq-iff ceiling-eq-iff)
  hence $(\sum n. \sum x(\text{real } n..<.. \text{real}(\text{Suc } n)). (f(\text{real}(\text{nat } [x]))) \partial \text{lborel}) = (\sum n. \sum x(\text{real } n..<.. \text{real}(\text{Suc } n)). (f(\text{real}(\text{nat } [x]))) \partial \text{lborel})$
  by (intro suminf-cong nn-integral-cong) (auto simp: indicator-def)
  also have $\ldots = (\sum x(\text{real } i..<.. \text{real}(\text{Suc } i))). (f(\text{real}(\text{nat } [x::real]))) \partial \text{lborel})$
  by (subst nn-integral-disjoint-family)
  (auto simp: disjoint-family-on-def)
  also have $\ldots \leq (\sum x(\text{real } i..<.. \text{real}(\text{Suc } i))). (f(\text{nat } [x::real])) \partial \text{lborel})$
  by (intro nn-integral-mono) (auto simp: indicator-def intro!: decreasing)

finally show $\exists \text{thesis }$
qed

lemma nn-integral-has-integral-lebesgue:
fixes $f :: 'a::euclidean-space \Rightarrow \text{real}$
assumes nonneg: $\forall x. x \in \Omega \Longrightarrow 0 \leq f x$ and $I: (f\text{has-integral } I) \Omega$
shows $\text{integral}^N \text{lborel } (\lambda x. \text{indicator } \Omega \ast f x) = I$
proof

from I have \((\lambda x. \text{indicator } \Omega \ x \star f \ x) \in \text{lebesgue }\to_M \text{borel}\)
  by (rule has-integral-implies-lebesgue-measurable)
then obtain \(f' \:: 'a \to \text{real}\)
  where [measurable]: \(f' \in \text{borel }\to_M \text{borel}\) and eq: \(AE \ x \text{ in } \text{borel. indicator } \Omega \\
  x \star f \ x = f' \ x\)
  by (auto dest: completion-ex-borel-measurable-real)

from I have \(((\lambda x. \text{abs } \text{(indicator } \Omega \ x \star f \ x)) \text{ has-integral } I) \text{ UNIV}\)
  using nonneg by (simp add: indicator-def if-distrib[of \(\lambda x. x \star f \ y\) for \(y\) cong: if-cong)
also have \(((\lambda x. \text{abs } \text{(indicator } \Omega \ x \star f \ x)) \text{ has-integral } I) \text{ UNIV }\to\text{ ((}\lambda x. \text{abs } \text{(}f'\text{ }x)) \text{ has-integral } I\text{ UNIV}\)
  using eq by (intro has-integral-AE) auto
finally have integral\(^N\) lborel \((\lambda x. \text{abs } \text{(}f'\text{ }x)) = I\)
  by (rule nn-integral-has-integral-lborel[rotated 2]) auto
also have integral\(^N\) lborel \((\lambda x. \text{abs } \text{(}f'\text{ }x)) = \text{integral}^N\ lborel \((\lambda x. \text{abs } \text{(indicator } \Omega \ x \star f \ x))\)
  using eq by (intro nn-integral-cong-AE) auto
also have \((\lambda x. \text{abs } \text{(indicator } \Omega \ x \star f \ x)) = (\lambda x. \text{indicator } \Omega \ x \star f \ x)\)
  using nonneg by (auto simp: indicator-def fun-eq-iff)
finally show ?thesis.

qed

lemma decreasing-sum-le-integral:
fixes f :: real \to real
assumes nonneg: \(\forall x. x \geq 0 \implies f x \geq 0\)
assumes decreasing: \(\forall x y. 0 \leq x \implies x \leq y \implies f y \leq f x\)
assumes integral: \((f \text{ has-integral } I) \{0..\}\)
shows summable \((\lambda i. f \text{ (real } (\text{Suc } i)))\) and suminf \((\lambda i. f \text{ (real } (\text{Suc } i)))\) \leq I

proof
  have [simp]: \(I = 0\)
    by (intro has-integral-nonneg[OF integral] nonneg) auto
  have \((\sum n. \text{ennreal } (f (\text{Suc } n))) = \\
    (\sum n. \int \text{+} x \in \text{ (real } n \ldots \text{real } (\text{Suc } n)) \cdot \text{ennreal } (f (\text{Suc } n)) \text{ } \partial \text{borel})\) \\
    by (subst nn-integral-cmult-indicator) auto
  also have \(\text{nat } [x] = \text{Suc } n\) if \(x \in \text{ (real } n \ldots \text{real } (\text{Suc } n))\) for \(x n\)
    using that by (auto simp: nat-eq-iff ceiling-eq-iff)
  hence \((\sum n. \int \text{+} x \in \text{ (real } n \ldots \text{real } (\text{Suc } n)) \cdot \text{ennreal } (f (\text{Suc } n)) \text{ } \partial \text{borel}) = \\
    (\sum n. \int \text{+} x \in \text{ (real } n \ldots \text{real } (\text{Suc } n)) \cdot \text{ennreal } (f \text{ (real } (\text{nat } [x]))) \text{ } \partial \text{borel})\) \\
    by (intro suminf-cong nn-integral-cong) (auto simp: indicator-def)
  also have \(\ldots = (\int \text{+} x \in \bigcup i. \text{ (real } i \ldots \text{real } (\text{Suc } i)) \cdot \text{ennreal } (f \text{ (real } (\text{nat } [x\ldots])) \text{ } \partial \text{borel})\)
    by (subst nn-integral-disjoint-family) \\
    (auto simp: disjoint-family-on-def intro!: measurable-completion)
  also have \(\ldots \leq (\int \text{+} x \in \{0..\} \cdot \text{ennreal } (f x) \text{ } \partial \text{borel})\) \\
    by (intro nn-integral-mono) (auto simp: indicator-def nonneg intro!: decreasing)
  also have \(\ldots = (\int \text{+} x. \text{ennreal } (\text{indicat-real } \{0..\} \ x \star f \ x) \text{ } \partial \text{borel})\) \\
    by (intro nn-integral-cong) (auto simp: indicator-def)
  also have \(\ldots = \text{ennreal } I\)
using nn-integral-has-integral-lebesgue[OF nonneg integral] by (auto simp: nonneg)
finally have *: (\(\sum n. \text{ennreal } (f \text{ (Suc } n))\)) \(\leq\) \text{ennreal } I .
from * show summable: summable (\(\lambda i. f \text{ (real } (\text{Suc } i))\))
  by (intro summable-suminf-not-top) (auto simp: top-unique intro: nonneg)
note *
also from summable have (\(\sum n. \text{ennreal } (f \text{ (Suc } n))\)) = \text{ennreal } (\(\sum n. f \text{ (Suc } n)\))
  by (subst suminf-ennreal2) (auto simp: o-def nonneg)
finally show (\(\sum n. f \text{ (real } (\text{Suc } n))\)) \(\leq\) I by (subst (asm) ennreal-le-iff) auto
qed

lemma decreasing-sum-le-integral':
  fixes f :: \(\text{real } \Rightarrow \text{real}\)
  assumes \(\forall x. x \geq 0 \implies f x \geq 0\)
  assumes \(\forall x y. 0 \leq x \implies x \leq y \implies f y \leq f x\)
  assumes \(f \text{ has-integral } I \{0..\}\)
  shows summable (\(\lambda i. f \text{ (real } i)\)) and suminf (\(\lambda i. f \text{ (real } i)\)) \(\leq\) \(f 0 + I\)
proof –
  have summable (\(\lambda x. f \text{ (real } (\text{Suc } i)))\))
    using decreasing-sum-le-integral[OF assms]
    by (simp add: o-def)
  thus *: summable (\(\lambda i. f \text{ (real } i)\)) by (subst (asm) summable-Suc-iff)
  have (\(\sum n. f \text{ (real } (\text{Suc } n))) \leq I\) by (intro decreasing-sum-le-integral assms)
  thus suminf (\(\lambda i. f \text{ (real } i)\)) \(\leq\) \(f 0 + I\)
    using * by (subst (asm) suminf-split-head) auto
qed

lemma norm-suminf-le:
  assumes \(\forall n. \text{norm } (f n :: 'a :: \text{banach}) \leq g n\) summable g
  shows \(\text{norm } (\text{suminf } f) \leq \text{suminf } g\)
proof –
  have *: summable (\(\lambda n. \text{norm } (f n))\)) using assms
    by (intro suminf-le assms allI)
  hence \(\text{norm } (\text{suminf } f) \leq (\sum n. \text{norm } (f n))\) by (intro summable-norm)
  also have \(\ldots \leq \text{suminf } g\) by (intro suminf-le * assms allI)
  finally show ?thesis .
qed

lemma of-nat-powr-neq-1-complex [simp]:
  assumes \(n > 1 \text{ Re } s \neq 0\)
  shows \(\text{of-nat } n \text{ powr } s \neq (1 :: \text{complex})\)
proof –
  have \(\text{norm } (\text{of-nat } n \text{ powr } s) = \text{real } n \text{ powr } \text{Re } s\)
    by (simp add: norm-powr-real-powr)
  also have \(\ldots \neq 1\)
    using assms by (auto simp: powr-def)
  finally show ?thesis by auto
qed
lemma abs-summable-on-uminus-iff:
  \((\lambda x. -f x)\) abs-summable-on \(A\) \iff \(f\) abs-summable-on \(A\)
using abs-summable-on-uminus[of \(f\) \(A\)] abs-summable-on-uminus[of \(\lambda x. -f x\) \(A\)]
by auto

lemma abs-summable-on-cmult-right-iff:
  fixes \(f\) :: \('a \Rightarrow \, 'b\)
  {\text{banach, real-normed-field, second-countable-topology}}
assumes \(c \neq 0\)
shows \((\lambda x. c \cdot f x)\) abs-summable-on \(A\) \iff \(f\) abs-summable-on \(A\)
using assms abs-summable-on-cmult-right[of \(c\) \(f\) \(A\)] abs-summable-on-cmult-right[of inverse \(c\) \(\lambda x. c \cdot f x\) \(A\)]
by (auto simp: field-simps)

lemma abs-summable-on-cmult-left-iff:
  fixes \(f\) :: \('a \Rightarrow \, 'b\)
  {\text{banach, real-normed-field, second-countable-topology}}
assumes \(c \neq 0\)
shows \((\lambda x. f x \cdot c)\) abs-summable-on \(A\) \iff \(f\) abs-summable-on \(A\)
using assms abs-summable-on-cmult-left[of \(c\) \(f\) \(A\)] abs-summable-on-cmult-left[of inverse \(c\) \((\lambda x. f x)\) \(A\)]
by (auto simp: field-simps)

lemma fds-logderiv-completely-multiplicative:
  fixes \(f\) :: \('a \Rightarrow \, \text{real-normed-field}\) \(fds\)
assumes completely-multiplicative-function \((fds-nth f)\) \(fds-nth f 1\) \(\neq 0\)
shows \(fds-deriv f / f = -fds (\lambda n. fds-nth f n \cdot \text{mangoldt}\ n)\)
proof
  have \(fds-deriv f / f = -fds (\lambda n. fds-nth f n \cdot \text{mangoldt}\ n) \cdot f / f\)
  using completely-multiplicative-fds-deriv[of \(fds-nth f\) assms] by simp
also have \(\ldots = -fds (\lambda n. fds-nth f n \cdot \text{mangoldt}\ n)\)
  using assms by (simp add: divide-fds-def fds-right-inverse)
finally show ?thesis .
qed

lemma fds-nth-logderiv-completely-multiplicative:
  fixes \(s\) :: \('a \Rightarrow \, \text{dirichlet-series}\) and \(l\) :: \('a\) and \(f\) :: \('a \Rightarrow \, \text{fds}\)
defines \(h\equiv fds-deriv f / f\)
assumes completely-multiplicative-function \((fds-nth f)\) \(fds-nth f 1\) \(\neq 0\)
shows \(fds-nth (fds-deriv f / f) n = -fds-nth f n \cdot \text{mangoldt}\ n\)
using assms by (subst fds-logderiv-completely-multiplicative) (simp-all add: fds-nth-fds')

lemma eval-fds-logderiv-completely-multiplicative:
  fixes \(s\) :: \('a \Rightarrow \, \text{dirichlet-series}\) and \(l\) :: \('a\) and \(f\) :: \('a \Rightarrow \, \text{fds}\)
defines \(h\equiv fds-deriv f / f\)
assumes completely-multiplicative-function \((fds-nth f)\) and \(fds-nth f 1\) \(\neq 0\)
assumes \(s \cdot 1 > \text{abs-conv-abscissa}\ f\)
shows \((\lambda p. \text{of-real} (\ln (\text{real}\ p)) \cdot (1 \cdot (1 - fds-nth f p \cdot \text{nat-power}\ p\ s) - 1))\)
  abs-summable-on \{\(p. \text{prime}\ p\)\} (is \?th1)\)
and eval-fds h s = \(-\sum a p \mid \text{prime} p. \text{of-real} (\ln (\text{real}\ p)) \cdot (1 - fds-nth f p \cdot \text{nat-power}\ p\ s) - 1))\)
proof –
let ?P = \{p::nat. prime p\}
interpret f: completely-multiplicative-function fds-nth f by fact
have fds-abs-converges h s
  using abs-core-absissca-completely-multiplicative-log-deriv[OF assms(2)] assms
  by (intro fds-abs-converges) auto
hence*: (\lambda n. fds-nth h n / nat-power n s) abs-summable-on UNIV
  by (auto simp: h-def fds-abs-converges-altdef)
note *
also have (\lambda n. fds-nth h n / nat-power n s) abs-summable-on UNIV \iff
  (\lambda x. -fds-nth f x * mangoldt x / nat-power x s) abs-summable-on Collect primepow
unfolding h-def using fds-nth-logderiv-completely-multiplicative[OF assms(2)]
by (intro abs-summable-on-cong-neutral) auto
also have ?this \iff
  (\lambda (p,k). -((fds-nth f p / nat-power p s)) ^ Suc k * of-real (ln (real p))) abs-summable-on (\?P \times UNIV)
unfolding case-prod-unfold
  by (intro abs-summable-on-cong, subst mangoldt-primepow)
also have \ldots \iff
  (\lambda (p,k). -(\lambda n. (norm (fds-nth f p / nat-power p s))) ^ Suc n) \if
  prime p for p
proof –
  have summable (\lambda n. (norm (fds-nth f p / nat-power p s))) ^ Suc n)
  \if prime p for p
unfolding abs-summable-on-nat-iff'
  by (simp add: norm-power norm-mult norm-divide mult_ac dest: prime-gt-1-nat)
thus ?thesis by (rule summable-mult-D) (insert p, auto dest: prime-gt-1-nat)
qed
have sums: (\lambda n. (fds-nth f p / nat-power p s) ^ Suc n) sums
  (1 / (1 - fds-nth f p / nat-power p s) - 1) \if p: prime p for p :: nat
proof –
  from sum4[OF p] have norm (fds-nth f p / nat-power p s) < 1
  unfolding summable-Suc-iff by (simp add: summable-geometric-iff)
  from geometric-sums[OF this] show ?thesis by (subt sums-Suc-iff) auto
qed
have eq: \((\sum_a k \cdot ((\text{fds-nth} \ f \ p \ / \ \text{nat-power} \ p \ s) \cdot \text{Suc} \ k \ * \ \text{of-real} \ (\text{ln} \ (\text{real} \ p))))\) 
\(- (\text{of-real} \ (\text{ln} \ (\text{real} \ p)) \ * (1 / (1 - \text{fds-nth} \ f \ p \ / \ \text{nat-power} \ p \ s) - 1)) \)
if \(p\): prime \(p\) for \(p\)

proof

have \((\sum_a k \cdot ((\text{fds-nth} \ f \ p \ / \ \text{nat-power} \ p \ s) \cdot \text{Suc} \ k) * \text{of-real} (\text{ln} \ (\text{real} \ p))\) 
using \(\text{sum}4[\text{of} \ p]\) 
by (subst \text{infsetsum-cmult-left} [\text{symmetric}])
(auto simp: \text{abs-summable-on-nat-iff} \ norm-power simp del: \text{power-Suc})

also have \((\sum_a k \cdot ((\text{fds-nth} \ f \ p \ / \ \text{nat-power} \ p \ s) \cdot \text{Suc} \ k) = 
(1 / (1 - \text{fds-nth} \ f \ p \ / \ \text{nat-power} \ p \ s) - 1) \) using \(\text{sum}4[\text{OF} \ p]\)

\text{sums}[\text{OF} \ p]

by (subst \text{infsetsum-nat})
(auto simp: \text{sums-iff} \text{abs-summable-on-nat-iff} \ norm-power simp del: \text{power-Suc})

finally show \(?\text{thesis}\) by (simp add: \text{mult-ac})

qed

have \(\text{sum}3\): \((\lambda x. \sum_a y \cdot ((\text{fds-nth} \ f \ x \ / \ \text{nat-power} \ x \ s) \cdot \text{Suc} \ y * \text{of-real} \ (\text{ln} \ (\text{real} \ x))\) 
abs-summable-on \{p, \ \text{prime} \ p\}
using \(\text{sum}2\) by (rule \text{abs-summable-on-Sigma-project1'})
also have \(?\text{this} \leftrightarrow (\lambda x. \text{abs-summable-on} \ (\text{Sigma-project1')})\) \ text{auto}

\text{prime} \ p)
by (intro \text{abs-summable-on-cong eq}) \text{auto}
also have \(\ldots \leftrightarrow \ ?\text{th1}\) by (subst \text{abs-summable-on-unminus-iff}) \text{auto}
finally show \(?\text{th1}\)

have \(\text{eval-fds} \ h \ s \ = \ (\sum_a n \cdot \text{fds-nth} \ h \ n \ / \ \text{nat-power} \ n \ s)\)
using \(\text{unfolding} \ \text{eval-fds-def}\) by (subst \text{infsetsum-nat'}) \text{auto}
also have \(\ldots \cdot (\sum_a n \in \{n. \ \text{primepow} \ n\}, \text{-fds-nth} \ f \ n \ * \ \text{mangoldt} \ n \ / \ \text{nat-power} \ n \ s)\)

\text{unfolding} \(h\)-def using \text{fds-nth-logderiv-completely-multiplicative}[\text{OF assms}(2)]
by (intro \text{infsetsum-cong-neutral}) \text{(auto simp: \text{fds-nth-fds mangoldt-def})}
also have \(\ldots \cdot (\sum_a (p,k) \in (\text{P} \times \text{UNIV}), \text{-fds-nth} \ f \ (p \ \text{Suc} \ k) * \text{mangoldt} \ (p \cdot \text{Suc} \ k) / \text{nat-power} \ (p \cdot \text{Suc} \ k) \ s)\)
using \text{bij-betw-primepow} unfolding case-prod-unfold
by (intro \text{infsetsum-reindex-bij-betw} [\text{symmetric}])
also have \(\ldots \cdot (\sum_a (p,k) \in (\text{P} \times \text{UNIV}). \text{-fds-nth} \ f \ (p \ \text{Suc} \ k) * \text{of-real} \ (\text{ln} \ (\text{real} \ p)))\)
by (intro \text{infsetsum-cong})
(auto simp: \text{f.mult f.power mangoldt-def aprimefactor-prime-power ln-realpow prime-gt-0-nat})

\text{nat-power-power-left divide-simps simp del: \text{power-Suc})}
also have \(\ldots \cdot (\sum_a p \ | \ \text{prime} \ p. \sum_a k.
= \ ((\text{fds-nth} \ f \ p \ / \ \text{nat-power} \ p \ s) \cdot \text{Suc} \ k) * \text{of-real} \ (\text{ln} \ (\text{real} \ p))))\)
lemma eval-fds-logderiv-zeta:
assumes Re s > 1
shows \((\lambda p. \text{of-real } (\ln (\text{real } p)) / (p \text{ powr } s - 1))\) \(\text{abs-summable-on } \{p. \text{prime } p\}\) (is ?th1)
and deriv zeta s / zeta s = 
\(- (\sum a p \mid \text{prime } p. \text{of-real } (\ln (\text{real } p)) / (p \text{ powr } s - 1))\) (is ?th2)

docthm
- have *: \(\text{completely-multiplicative-function} (\text{fds-nth fds-zeta} :: \cdot \Rightarrow \text{complex})\)
  by standard auto
- note abscissa = le-less-trans[OF abs-conv-abscissa-completely-multiplicative-log-deriv[OF *
  ]] have \((\lambda p. \ln (\text{real } p) * (1 / (1 - \text{fds-nth fds-zeta } p / p \text{ powr } s - 1))\) \(\text{abs-summable-on } \{p. \text{prime } p\}\)
  using eval-fds-logderiv-completely-multiplicative[OF *, of s] assms by auto
- also have ?this \(\Rightarrow (\lambda p. \ln (\text{real } p) / (p \text{ powr } s - 1))\) \(\text{abs-summable-on } \{p. \text{prime } p\}\) using assms
  by (intro abs-summable-on-cong) (auto simp: fds-nth-zeta divide-simps dest: prime-gt-1-nat)
- finally show ?th1 .

from assms have ev: eventually \((\lambda z. \ z \in \{z. \ Re z > 1\})\) (nhds s)
  by (intro eventually-nhds-in-open open-halfspace-Real-gt) auto
- have deriv zeta s = deriv (eval-fds fds-zeta) s
  by (intro deriv-cong-ev[OF eventually-mono[OF ev]])) (auto simp: eval-fds-zeta)
- also have deriv (eval-fds fds-zeta) s / zeta s = eval-fds (fds-deriv fds-zeta / fds-zeta) s
  using assms zeta-Real-gt-1-nonzero[of s]
  by (subt eval-fds-log-deriv) (auto simp: eval-fds-zeta eval-fds-deriv intro!: abscissa)
- also have eval-fds (fds-deriv fds-zeta / fds-zeta) s = 
  \(- (\sum a p \mid \text{prime } p. \ln (\text{real } p) * (1 / (1 - \text{fds-nth fds-zeta } p / p \text{ powr } s - 1))\)
  (is - = ?S) using eval-fds-logderiv-completely-multiplicative[OF *, of s] assms
  by auto
- also have ?S = \((\sum a p \mid \text{prime } p. \ln (\text{real } p) / (p \text{ powr } s - 1))\) using assms
  by (intro infsetsum-cong) (auto simp: fds-nth-zeta divide-simps dest: prime-gt-1-nat)
- finally show ?th2 .

docthm

lemma sums-logderiv-zeta:
assumes Re s > 1
shows \((\lambda p. \text{if prime } p \text{ then of-real } (\ln (\text{real } p)) / (\text{of-nat } p \text{ powr } s - 1) \text{ else 0})\)
sums 
\[ -(\text{deriv zeta } s) / \text{zeta } s) \text{ (is {?} sums -) } \]

proof –

note * = eval-fds-logderiv-zeta[OF assms]

from sums-infsetsum-nat[OF *1(1)] and *2 show ?thesis by simp

qed

lemma abs-conv-abscissa-diff-le:
\[ \text{abs-conv-abscissa } (f - g : \text{dirichlet-series fds}) \leq \]

using abs-conv-abscissa-add-le[of f - g] by (auto simp:)

lemma abs-conv-abscissa-diff-leI:
\[ \text{abs-conv-abscissa } (f : \text{dirichlet-series fds}) \leq d \implies \text{abs-conv-abscissa } g \leq d \]

using abs-conv-abscissa-diff-le[of f g] by (auto simp:)

lemma range-add-nat:
\[ \text{range } (\lambda n. n + c) = \{c :: \text{nat} ..\} \]

proof safe

fix x assume x ≥ c

hence x = x - c + c by simp

thus x ∈ range (λ n. n + c) by blast

qed auto

lemma abs-summable-hurwitz-zeta:
\[ \text{assumes Re } s > 1 \text{ a + real b > 0} \]

shows \[ (λ n. 1 / (\text{of-nat } n + a) \text{ powr } s) \text{ abs-summable-on } \{b ..\} \]

proof –

from assms have summmable (λ n. cmod (1 / (of-nat (n + b) + a) powr s))

using summable-hurwitz-zeta-real[of Re s a + b]

by (auto simp: norm-divide powr-minus field-simps norm-powr-real-powr)

hence (λ n. 1 / (of-nat (n + b) + a) powr s) abs-summable-on UNIV

by (auto simp: abs-summable-on-nat-iff add-ac)

also have ?this ↵ (λ n. 1 / (of-nat n + a) powr s) abs-summable-on range

(λ n. n + b)

by (rule abs-summable-on-reindex-iff auto)

also have range (λ n. n + b) = {b ..} by (rule range-add-nat)

finally show ?thesis .

qed

lemma hurwitz-zeta-nat-conv-infsetsum:
\[ \text{assumes a > 0 and Re } s > 1 \]

shows \[ \text{hurwitz-zeta } (\text{real a}) s = (\sum a n. \text{of-nat } n + a) \text{ powr } -s \]

hurwitz-zeta (real a) s = (\sum a n ∈ [a..]. of-nat n powr -s)

proof –

have hurwitz-zeta (real a) s = (\sum n. of-nat (n + a) powr -s)

using assms by (subst hurwitz-zeta-cone-suminf) auto

also have . . . = (\sum a n. of-nat (n + a) powr -s)
lemma continuous-pre-zeta [continuous-intros]:
assumes 0 < Re s and a: a > 0
shows norm (pre-zeta a s) ≤ (1 + norm s / Re s) / 2 * a powr − Re s
proof –
let ?f = λx. − (s * (x + a) powr (−1 − s))
let ?g' = λx. norm s * (x + a) powr (−1 − Re s)
define R where R = EM-remainder 1 ?f 0
have [simp]: − Re s − 1 = −1 − Re s by (simp add: algebra-simps)

have frac x − 1 / 2 ≤ 1 / 2 for x :: real unfolding frac-def
by linarith

hence |bernpoly (Suc 0) x| ≤ 1 / 2 for x
by (simp add: bernpoly-def bernpoly-def)

moreover have ((λb. cmod s * (b + a) powr − Re s / Re s) ----> 0) at-top
using Re s > 0) (a > 0) by real-asymp

ultimately have *: ∀ x. x ≥ real 0 ----> norm (EM-remainder 1 ?f (int x)) ≤ (1 / 2) / fact 1 * (−?g (real x))
using a > 0) (Re s > 0)
by (intro norm-EM-remainder-le-strong-nat"where g' = ?g' and Y = {\{}")
(auto intro!: continuous-intros derivative-eq-intros)
simp: field-simps norm-mult norm-powr-real-powr add-eq-0-iff)

have R: norm R \leq norm s / (2 * Re s) * a powr -Re s
  unfolding R-def using spec[OF *, of 0] by simp

from assms have pre-zeta a s = a powr -s / 2 + R
  by (simp add: pre-zeta-def pre-zeta-aux-def R-def)
also have norm ... \leq a powr -Re s / 2 + norm s / (2 * Re s) * a powr -Re
s using a
  by (intro order.trans[OF norm-triangle-ineq add-mono R])
also have ... = (1 + norm s / Re s) / 2 * a powr -Re s
  by (simp add: field-simps)
finally show thesis .

qed

lemma pre-zeta-bound':
  assumes 0 < Re s and a: a > 0
  shows norm (pre-zeta a s) \leq norm s / (Re s * a powr Re s)
proof –
  from assms have norm (pre-zeta a s) \leq (1 + norm s / Re s) / 2 * a powr -Re
s
  by (intro pre-zeta-bound) auto
also have ... = (Re s + norm s) / 2 / (Re s * a powr Re s)
  by auto simp: norm-powr-real-powr
also have Re s + norm s \leq norm s + norm s by (intro add-right_mono)
also have (norm s + norm s) / 2 = norm s by simp
finally show norm (pre-zeta a s) \leq norm s / (Re s * a powr Re s)
  using assms by (simp add: divide-right_mono)

qed

lemma summable-comparison-test-bigo:
  fixes f :: nat \Rightarrow real
  assumes summable (\lambda n. norm (g n)) f \epsilon O(g)
  shows summable f
proof –
  from f \epsilon O(g) obtain C where C: eventually (\lambda x. norm (f x) \leq C * norm
(g x)) at-top
  by (auto elim: eventually_mono)
  thus thesis
  by (rule summable-comparison-test-ev) (insert assms, auto intro: summable-mult)

qed

lemma deriv-zeta-eq:
  assumes s: s \neq 1
  shows deriv zeta s = deriv (pre-zeta 1) s - 1 / (s - 1)^2
proof –
  from s have ev: eventually (\lambda z. z \neq 1) (nhds s) by (intro t1-space-nhds)
  have [derivative-intros]: (pre-zeta 1 has-field-derivative deriv (pre-zeta 1) s) (at
s)
by (intro holomorphic-derivI[of - UNIV] holomorphic-intros) auto
have \((\text{As. pre-zeta 1 } s + 1 / (s - 1)) \text{ has-field-derivative} \)
\((\text{deriv (pre-zeta 1) } s - 1 / (s - 1)^2) \) (at \(s\))
using \(s\) by (auto intro: derivative-eq-intros simp: power2-eq-square)
also have \(?this \iff (\text{zeta has-field-derivative} \text{ (deriv (pre-zeta 1) } s - 1 / (s - 1)^2)) \) (at \(s\))
by (intro has-field-derivative-cong-ev eventually-mono[of ev])
(auto simp: zeta-def hurwitz-zeta-def)
finally show \(?thesis\) by (rule DERIV-imp-deriv)
qed

lemma zeta-remove-zero:
assumes \(\text{Re } s \geq 1\)
shows \((s - 1) \ast \text{pre-zeta 1 } s + 1 \neq 0\)
proof (cases \(s = 1\))
case False
hence \((s - 1) \ast \text{pre-zeta 1 } s + 1 = (s - 1) \ast \text{zeta } s\)
by (simp add: zeta-def hurwitz-zeta-def divide-simps)
also from False assms have \(\ldots \neq 0\) using zeta-Re-ge-1-nonzero[of \(s\)] by auto
finally show \(?thesis\).
qed auto

lemma eval-fds-deriv-zeta:
assumes \(\text{Re } s > 1\)
shows \(\text{eval-fds (fds-deriv fds-zeta) } s = \text{deriv zeta } s\)
proof
have ev: \(\text{eventually } (\lambda z. z \in \{z. \text{Re } z > 1\}) \) (nhds \(s\))
using assms by (auto simp add: sum-upto-altdef nat-add-distrib)
from assms have \(\text{eval-fds (fds-deriv fds-zeta) } s = \text{deriv (eval-fds fds-zeta) } s\)
by (subst eval-fds-deriv) auto
also have \(\ldots = \text{deriv zeta } s\)
by (intro deriv-cong-ev eventually-mono[of ev]) (auto simp: eval-fds-zeta)
finally show \(?thesis\).
qed

lemma length-sorted-list-of-set [simp]:
\(\text{finite } A \implies \text{length (sorted-list-of-set } A) = \text{card } A\)
by (metis length-remdups-card-cone length-sort set-sorted-list-of-set
sorted-list-of-set-sort-remdups)

lemma le-nat-iff': \(x \leq \text{nat } y \iff x = 0 \land y \leq 0 \lor \text{int } x \leq y\)
by auto

lemma sum-upto-plus1:
assumes \(x \geq 0\)
shows \(\text{sum-upto } f (x + 1) = \text{sum-upto } f x + f (\text{Suc } (\text{nat } \lfloor x \rfloor))\)
proof
have \(\text{sum-upto } f (x + 1) = \text{sum } f \{0<..\text{Suc } (\text{nat } \lfloor x \rfloor)\}\)
using assms by (simp add: sum-upto-altdef nat-add-distrib)
also have $\{0 < \ldots \text{Suc} \ (\text{nat} \ [x])\} = \text{insert} \ (\text{Suc} \ (\text{nat} \ [x])) \ \{0 < \ldots \text{nat} \ [x]\}$
  by auto
also have $\sum f \ldots = \sum\text{upto} \ f \ x + f \ (\text{Suc} \ (\text{nat} \ [x]))$
  by (subst sum.insert; (auto simp: sum-upto-altdef add-ac)
finally show ?thesis .
qed

lemma sum-upto-minus1:
  assumes $x \geq 1$
  shows $\sum\text{upto} \ f \ (x - 1) = (\sum\text{upto} \ f \ x - f \ (\text{nat} \ [x]) :: \ 'a :: \ ab\text{-group-add})$
using sum-upto-plus1 [of $x - 1$ $f$] assms by (simp add: algebra_simps nat_diff_distrib)

lemma integral-small:
  fixes $f$ $g' :: \ real \Rightarrow \ real$
  assumes $f \in \text{o}(g')$ and $\text{filterlim} \ g \ \text{at-top} \ \text{at-top}$
  assumes $\forall a. \ a \leq a' \Longrightarrow a' \leq x \Longrightarrow f\text{ integrable-on} \ \{a' \ldots \}$
  assumes deriv: $\forall x. \ x \geq a \Longrightarrow (g \ \text{has-field-derivative} \ g' \ x) \ (\text{at} \ x)$
  assumes cont: $\text{continuous-on} \ \{a \ldots \} \ g'$
  assumes nonneg: $\forall x. \ x \geq a \Longrightarrow g' \ x \geq 0$
  shows $(\lambda x. \ \int \{a \ldots \} \ f) \in \text{o}(g)$
proof (rule landau-o.small)
  fix $c :: \ real$ assume $c > 0$
  note [continuous-intros] = continuous-on-subset[OF cont]
  define $c' \ where \ c' = c / 2$
  from $c$ have $c' :: \ c' > 0$ by (simp add: $c'$-def)
  from landau-o.smallD[OF assms(1) this] obtain $b \ where \ b : \ \forall x. \ x \geq b \Longrightarrow \ \int f \leq c' * \ \int (g' \ x)$
    unfolding eventually-at-top-linorder by blast
  define $b' \ where \ b' = \max a \ b$
  define $D \ where \ D = \text{norm} \ (\int \{a \ldots \} \ f)$
  have $\text{filterlim} \ (\lambda x. \ c' * \ g \ x) \ \text{at-top} \ \text{at-top}$
    using $c'$ by (intro filterlim-tends-to-pos-mult-at-top[OF tendsto-const] assms)
  hence eventually $(\lambda x. \ c' * \ g \ x \geq D - c' * g \ b') \ \text{at-top}$
    by (auto simp: filterlim-at-top)
  thus eventually $(\lambda x. \ \text{norm} \ (\int \{a \ldots \} \ f) \leq c * \ \text{norm} \ (g \ x)) \ \text{at-top}$
    using eventually-ge-at-top[of $b'$]
proof eventually-elim
  case (elim $x$)
  have $b' : \ a \leq b' \leq b'$ by (auto simp: $b'$-def)
  from elim $b'$ have integrable: $(\lambda x. \ g' \ x) \ \text{integrable-on} \ \{b' \ldots \}$
    by (intro integrable-continuous-real continuous-intros) auto
  have $(\int \{a \ldots \} \ f = \int \{a \ldots \} \ f + \int \{b' \ldots \} \ f)$
    using elim $b'$ by (intro integral-combine [symmetric] assms) auto
  also have $\text{norm} \ldots \leq D + \text{norm} \ (\int \{b' \ldots \})$
    unfolding $D$-def by (rule norm-triangle-ineq)
  also have $\text{norm} \ (\int \{b' \ldots \} \ f) \leq \int \{b' \ldots \} \ (\lambda x. \ c' * \ \text{norm} \ (g' \ x))$
    using $b'$ elim assms $c'$ integrable by (intro integral-norm-bound-integral $b$
      assms) auto

38
also have \( \ldots = c' * \text{integral} \{ b'..x \} \; (\lambda x. |g' x|) \) by simp
also have \( \text{integral} \{ b'..x \} \; (\lambda x. |g' x|) = \text{integral} \{ b'..x \} \; g' \)
using assms b' by (intro integral-cong) auto
also have \( (g' \; \text{has-integral} \; (g \; x - g \; b')) \{ b'..x \} \) using b' elim
by (intro fundamental-theorem-of-calculus)
(auto simp flip; has-field-derivative-iff-has-vector-derivative
intro!; has-field-derivative-at-within[OF deriv])
hence \( \text{integral} \{ b'..x \} \; g' = g \; x - g \; b' \)
by (simp add: has-integral-iff)
also have \( D + c' * (g \; x - g \; b') \leq c * g \; x \)
using elim by (simp add: field-simps c'-def)
also have \( \ldots \leq c * \text{norm} \; (g \; x) \)
using c by (intro mult-left-mono) auto
finally show ?case by simp
qed

lemma integral-bigo:
fixes \( f \; g \; g' : \text{real} \Rightarrow \text{real} \)
assumes \( f \in O(g') \) and \( \text{filterlim} \; g \; \text{at-top} \) \( \text{at-top} \)
assumes \( \forall \lambda a' \; x, \; a \leq a' \Rightarrow a' \leq x \Rightarrow f \; \text{integrable-on} \; \{ a'..x \} \)
assumes \( \text{deriv}: \forall \lambda x. \; x \geq a \Rightarrow (g \; \text{has-field-derivative} \; g' \; x) \; (\text{at} \; \text{x} \; \text{within} \; \{ a.. \}) \)
assumes \( \text{cont:} \; \text{continuous-on} \; \{ a.. \} \; g' \)
assumes \( \text{nonneg}: \forall \lambda x. \; x \geq a \Rightarrow g' \; x \geq 0 \)
shows \( (\lambda x. \; \text{integral} \; \{ a..x \} \; f) \in O(g) \)
proof
- note \( \text{continuous-intros} = \text{continuous-on-subset}[\text{OF cont}] \)
from \( \text{landau-o.bigE}[\text{OF assms(1)}] \)
obtain \( c \; b \) where \( c : \text{c} > 0 \) and \( b: \; \forall \lambda x. \; x \geq b \Rightarrow \text{norm} \; (f \; x) \leq c * \text{norm} \; (g' \; x) \)
unfolding \( \text{eventually-at-top-limorder} \) by metis
define \( c' \) where \( c' = c / 2 \)
define \( b' \) where \( b' = \text{max} \; a \; b \)
define \( D \) where \( D = \text{norm} \; (\text{integral} \; \{ a..b' \} \; f) \)
have \( \text{filterlim} \; (\lambda x. \; c * g \; x) \; \text{at-top} \; \text{at-top} \)
using \( c \) by (intro \( \text{filterlim-tendsto-pos-mult-at-top}[\text{OF tendsto-const}] \) \( \text{assms} \))
hence \( \text{eventually} \; (\lambda x. \; c * g \; x \geq D - c * g \; b') \; \text{at-top} \)
by \( \text{auto simp: \text{filterlim-at-top}} \)
hence \( \text{eventually} \; (\lambda x. \; \text{norm} \; (\text{integral} \; \{ a..x \} \; f) \leq 2 * c * \text{norm} \; (g \; x)) \; \text{at-top} \)
using \( \text{eventually-ge-at-top}[\text{OF \text{b'}}] \)
proof \( \text{eventually-elim} \)
-case \( \text{elim} \; x \)
have \( b': a \leq b' \leq b' \) by \( \text{auto simp: \text{b'}-def} \)
from \( \text{elim} \; b' \) have \( \text{integrable:} \; (\lambda x. \; |g' \; x|) \; \text{integrable-on} \; \{ b'..x \} \)
by \( \text{intro \text{integrable-continuous-real continuous-intros}} \) auto
have \( \text{integral} \; \{ a..x \} \; f = \text{integral} \; \{ a..b' \} \; f \; + \; \text{integral} \; \{ b'..x \} \; f \)
using \( \text{elim} \; b' \) by \( \text{intro \text{integral-combine \[\text{symmetric} \] \text{assms}}\) auto
also have \( \text{norm} \; \ldots \leq D + \text{norm} \; (\text{integral} \; \{ b'..x \} \; f) \)
unfolding \( D\text{-def} \) by (rule norm-triangle-ineq)
also have \( \text{norm} (\text{integral} \{b'.x\} f) \leq \text{integral} \{b'.x\} (\lambda x. c * \text{norm} (g' x)) \)
using \( b' \) elim \assms \c \text{ integrable} \text{ by} \ (\text{intro integral-norm-bound-integral} b \assms) \text{ auto} 
also have \( \ldots = c * \text{integral} \{b'.x\} (\lambda x. |g' x|) \) by simp
also have \( \text{integral} \{b'.x\} (\lambda x. |g' x|) = \text{integral} \{b'.x\} g' \)
using \assms \( b' \) by (intro integral-cong) auto
also have \( (g' \text{ has-integral} (g x - g b')) \{b'.x\} \) using \( b' \) elim
by (intro fundamental-theorem-of-calculus)
\( \text{(auto simp flip: has-field-derivative-iff-has-vector-derivative} \)
\( \text{intro: DERIV-subset(OF deriv)} \)
\( \text{hence} \ \text{integral} \{b'.x\} g' = g x - g b' \)
by (simp add: has-integral-iff)
also have \( D + c * (g x - g b') \leq 2 * c * g x \)
using elim by (simp add: field-simps c' -def)
also have \( \ldots \leq 2 * c * \text{norm} (g x) \)
using \( c \) by (intro mult-left-mono) auto
finally show \( ?\text{thesis} \) by simp
qed
thus \( ?\text{thesis} \) by (rule bigoI)
qed

**lemma** primepows-le-subset:
assumes \( x: x > 0 \) and \( l: l > 0 \)
shows \( \{(p, i). \text{prime} p \land l \leq i \land \text{real} (p ^ i) \leq x\} \subseteq \{\text{..nat } \lfloor \text{root} l x \rfloor\} \times \{\text{..nat } \lfloor \text{log} 2 x \rfloor\} \)
**proof** safe
fix \( i::\text{nat} \) assume \( p: \text{prime} p \geq l \text{ real} (p ^ i) \leq x \)
have \( \text{real} p ^ i \leq \text{real} p ^ l \) using \( pi x l \)
by (intro power-increasing) (auto dest: prime-gt-0-nat)
also have \( \ldots \leq x \) using \( pi \) by simp
finally have \( \text{root} l \text{ (real} p ^ l) \leq \text{root} l x \)
using \( x pi l \) by (subst real-root-le-iff) auto
also have \( \text{root} l \text{ (real} p ^ l) = \text{real} p \)
using \( pi l \) by (subst real-root-pos2) auto
finally show \( p \leq \text{nat } \lfloor \text{root} l x \rfloor \) using \( pi l x \) by (simp add: le-nat-iff' le-floor-iff)

from \( pi \) have \( 2 ^ i \leq \text{real} p ^ i \) using \( l \)
by (intro power-mono) (auto dest: prime-gt-1-nat)
also have \( \ldots \leq x \) using \( pi \) by simp
finally show \( i \leq \text{nat } \lfloor \text{log} 2 x \rfloor \) using \( pi x \)
by (auto simp: le-nat-iff' le-floor-iff le-log-iff powr-realpow)
qed

**lemma** mangoldt-non-primepow: \( \neg \text{primepow} n \implies \text{mangoldt} n = 0 \)
by (auto simp: mangoldt-def)

**lemma** le-imp-bigo-real:
assumes \( c \geq 0 \) eventually \((\lambda x. f x \leq c * (g x :: \text{real}))\) \( F \) eventually \((\lambda x. 0 \leq f \)

40
x) \( F \)
  shows \( f \in O[F](g) \)
proof  
  have eventually \((\lambda x. \text{norm}(f x) \leq c \ast \text{norm}(g x)) \) \( F \)
  using assms(2,3)
proof eventually-elim
  case (elim \( x \))
  have \( \text{norm}(f x) \leq c \ast g x \) using elim by simp
  also have \( \ldots \leq c \ast \text{norm}(g x) \) by (intro mult-left-mono assms) auto
finally show \(?case .
qed
thus \(?thesis by (intro bigoI[of - c]) auto
qed

lemma ln-minus-ln-floor-bigo: \((\lambda x. \ln x - \ln(\text{real} \lfloor x \rfloor)) \in O(\lambda - 1)\)
proof (intro le-imp-bigo-real[of 1] eventually-mono[of eventually-ge-at-top[of 1]])
fix \( x :: \text{real} \)
assume \( x : x \geq 1 \)
from \( x \) have \(*: x - \text{real}(\text{nat} \lfloor x \rfloor) \leq 1 \) by linarith
from \( x \) have \( \ln x - \ln(\text{real} \lfloor x \rfloor) \leq (x - \text{real}(\text{nat} \lfloor x \rfloor)) / \text{real}(\text{nat} \lfloor x \rfloor) \)
  by (intro ln-diff-le auto
also have \( \ldots \leq 1 / 1 \) using \( x \ast \) by (intro frac-le auto
finally show \( \ln x - \ln(\text{real}(\text{nat} \lfloor x \rfloor)) \leq 1 \ast 1 \) by simp
qed auto

lemma cos-geD:
  assumes \( \cos x \geq \cos a \leq a 0 \leq a a \leq \pi \)
  shows \( x \in \{-a..a\} \)
proof (cases \( x \geq 0 \))
  case True
  with assms show \(?thesis
    by (subst (asm) cos-mono-le-eq) auto
next
  case False
  with assms show \(?thesis using cos-mono-le-eq[of a -x] \)
    by auto
qed

lemma path-image-part-circlepath-same-Re:
  assumes \( 0 \leq b b \leq \pi a = -b r \geq 0 \)
  shows \( \text{path-image}(\text{part-circlepath } c r a b) = \text{sphere } c r \cap \{s. \text{Re } s \geq \text{Re } c + r \ast \cos a\} \)
proof safe
  fix \( z \) assume \( z \in \text{path-image}(\text{part-circlepath } c r a b) \)
  with assms obtain \( t \) where \( t : t \in \{a..b\} z = c + \text{of-real} r \ast \text{cis} t \)
    by (auto simp: path-image-part-circlepath exp-eq-polar)
from \( t \) and assms show \( z \in \text{sphere } c r \)
    by (auto simp: dist-norm norm-mult)
from \( t \) and \( \texttt{assms} \) show \( Re \ z \geq Re \ c + r * \cos a \)

using \( \texttt{cos-monotone-0-pi-le[0 \ t \ b]} \) \( \texttt{cos-monotone-minus-pi-0[0 \ a \ t]} \)
by (cases \( t \geq 0 \)) (auto intro!: \( \texttt{mult-left-mono} \))

next

fix \( z \) assume \( z: \in \text{sphere} \; c \; r \) \( \text{Re} \; z \geq \text{Re} \; c + r * \cos a \)

show \( z \in \text{path-image} \; (\text{part-circlepath} \; c \; r \; a \; b) \)
proof (cases \( r = 0 \))

case False

with \( \texttt{assms} \) have \( r: \; r > 0 \) by simp

with \( z \) have \( \text{z-eq:} \; z = c + r * \text{cis} \; (\text{Arg} \; (z - c)) \)

using \( \texttt{Arg-eq[of z - c]} \) by (auto simp: \( \texttt{dist-norm \; exp-eq-polar \; norm-minus-commute} \))

moreover from \( z(2) \) \( r \) \( \texttt{assms} \) have \( \cos b \leq \cos (\text{Arg} \; (z - c)) \)

by (subst \( \texttt{asm(z-eq)} \)) auto

with \( \texttt{assms} \) have \( \text{Arg} \; (z - c) \in \{-b..b\} \)

using \( \texttt{Arg-le-pi[of z - c]} \) \( \texttt{mpi-less-Arg[of z - c]} \) by (intro \( \texttt{cos-geD} \)) auto

ultimately show \( z \in \text{path-image} \; (\text{part-circlepath} \; c \; r \; a \; b) \)

using \( \texttt{assms} \) by (subst \( \texttt{path-image-part-circlepath} \)) (auto simp: \( \texttt{exp-eq-polar} \))

qed (insert \( \texttt{assms \ z, auto \; simp: \; path-image-part-circlepath} \))

qed

lemma \( \text{part-circlepath-rotate-left} \):
\( \text{part-circlepath} \; c \; r \; (x + a) \; (x + b) = (\lambda z. \; c + \text{cis} \; x * (z - c)) \circ \text{part-circlepath} \; c \; r \; a \; b \)

by (simp add: \( \text{part-circlepath-def \; exp-eq-polar \; fun-eq-iff} \)
linepath-translate-left linepath-translate-right cis-mult add-ac)

lemma \( \text{part-circlepath-rotate-right} \):
\( \text{part-circlepath} \; c \; r \; (a + x) \; (b + x) = (\lambda z. \; c + \text{cis} \; x * (z - c)) \circ \text{part-circlepath} \; c \; r \; a \; b \)

by (simp add: \( \text{part-circlepath-def \; exp-eq-polar \; fun-eq-iff} \)
linepath-translate-left linepath-translate-right cis-mult add-ac)

lemma \( \text{path-image-semicircle-Re-ge} \):

assumes \( r \geq 0 \)

shows \( \text{path-image} \; (\text{part-circlepath} \; c \; r \; \{-\pi/2\} \; \{\pi/2\}) = \text{sphere} \; c \; r \cap \{s. \; \text{Re} \; s \geq \text{Re} \; c\} \)

by (subst \( \text{path-image-part-circlepath-same-Re} \)) (simp-all add: \( \texttt{assms} \))

lemma \( \text{sphere-rotate} \): \( (\lambda z. \; c + \text{cis} \; x * (z - c)) \; \circ \; \text{sphere} \; c \; r = \text{sphere} \; c \; r \)

proof safe

fix \( z \) assume \( z: \; z \in \text{sphere} \; c \; r \)

hence \( z = c + \text{cis} \; x * (c + \text{cis} \; (-x) * (z - c) - c) \)

\( c + \text{cis} \; (-x) * (z - c) \in \text{sphere} \; c \; r \)

by (auto simp: \( \texttt{dist-norm \; norm-mult \; norm-minus-commute} \)
cis-conv-exp \( \texttt{exp-minus \; field-simps \; norm-divide} \))

with \( z \) show \( z \in (\lambda z. \; c + \text{cis} \; x * (z - c)) \; \circ \; \text{sphere} \; c \; r \) by blast

qed (auto simp: \( \texttt{dist-norm \; norm-minus-commute \; norm-mult} \))
lemma path-image-semicircle-Re-le:
assumes $r \geq 0$
shows $\text{path-image} (\text{part-circlepath} \ c \ r \ (\pi/2) \ (3/2*\pi)) = \text{sphere} \ c \ r \ \cap \ {s. \ Re \ s \leq \ Re \ c}$
proof
let $\mathcal{f} = (\lambda z. \ c + \text{cis} \ (\pi/2) * (z - c))$
have $\ast: \text{part-circlepath} \ c \ r \ (\pi/2) \ (3/2*\pi) = \text{part-circlepath} \ c \ r \ (\pi + (-\pi/2))$
\begin{itemize}
\item by simp
\end{itemize}
have $\text{path-image} (\text{part-circlepath} \ c \ r \ (\pi/2) \ (3/2*\pi)) = \mathcal{f} \ \cap \ \mathcal{f} \ \cap \ \{s. \ Re \ c \leq \ Re \ s\}$
unfolding $\ast \ \text{part-circlepath-compose} \ \text{path-image-compose} \ \text{path-image-semicircle-Re-ge}[OF \ \text{assms}]$
\begin{itemize}
\item by auto
\end{itemize}
also have $\mathcal{f} \ \text{sphere} \ c \ r = \text{sphere} \ c \ r$
by (rule sphere-rotate)
also have $\mathcal{f} \ \{s. \ Re \ c \leq \ Re \ s\} = \{s. \ Re \ c \geq \ Re \ s\}$
by (auto simp: image-iff intro: exI[of - 2 * c - x for x])
finally show $\ast$.
qed

lemma path-image-semicircle-Re-ge:
assumes $r \geq 0$
shows $\text{path-image} (\text{part-circlepath} \ c \ r \ 0 \ \pi) = \text{sphere} \ c \ r \ \cap \ {s. \ Im \ s \geq \ Im \ c}$
proof
let $\mathcal{f} = (\lambda z. \ c + \text{cis} \ (\pi/2) * (z - c))$
have $\ast: \text{part-circlepath} \ c \ r \ 0 \ \pi = \text{part-circlepath} \ c \ r \ (\pi / 2 + (-\pi/2)) \ (\pi / 2)
+ \pi/2)$
\begin{itemize}
\item by simp
\end{itemize}
have $\text{path-image} (\text{part-circlepath} \ c \ r \ 0 \ \pi) = \mathcal{f} \ \cap \ \mathcal{f} \ \cap \ \{s. \ Re \ c \leq \ Re \ s\}$
unfolding $\ast \ \text{part-circlepath-compose} \ \text{path-image-compose} \ \text{path-image-semicircle-Re-ge}[OF \ \text{assms}]$
\begin{itemize}
\item by auto
\end{itemize}
also have $\mathcal{f} \ \text{sphere} \ c \ r = \text{sphere} \ c \ r$
by (rule sphere-rotate)
also have $\mathcal{f} \ \{s. \ Re \ c \leq \ Re \ s\} = \{s. \ Im \ c \leq \ Im \ s\}$
by (auto simp: image-iff intro: exI[of - c - i * (x - c) for x])
finally show $\ast$.
qed

lemma path-image-semicircle-Im-le:
assumes $r \geq 0$
shows $\text{path-image} (\text{part-circlepath} \ c \ r \ \pi \ (2 * \pi)) = \text{sphere} \ c \ r \ \cap \ {s. \ Im \ s \leq \ Im \ c}$
proof
let $\mathcal{f} = (\lambda z. \ c + \text{cis} \ (3*\pi/2) * (z - c))$
have $\ast: \text{part-circlepath} \ c \ r \ (2*\pi) = \text{part-circlepath} \ c \ r \ (3*\pi/2 + (-\pi/2))$
\[(3\pi/2 + \pi/2)\]

by simp

have path-image (part-circlepath c r (2 \* \pi)) = ?f' sphere c r \cap \{s. Re c \leq Re s\}

unfolding * part-circlepath-rotate-left path-image-compose path-image-semicircle-Re-ge [OF assms]

by auto

also have ?f' sphere c r = sphere c r

by (rule sphere-rotate)

also have \(\text{cis} (3 \pi / 2) = -i\)

using cis-mult [of \(\pi \pi / 2\)] by simp

hence \(\{s. \text{Re} c \leq \text{Re} s\}\) = \(\{s. \text{Im} c \geq \text{Im} s\}\)

by (auto simp: image-iff intro !: exI [of \(- c + i * (x - c)\) for \(x\)])

finally show ?thesis.

qed

lemma pour-numeral [simp]: \(x \geq 0 \Rightarrow (x::real) \text{pour numeral} y = x \cdot \text{numeral} y\)

using pour-numeral[of \(x\) \(y\)] by (cases \(x\) = 0) auto

lemma eval-fds-logderiv-zeta-real:

assumes \(x > (1 :: real)\)

shows \((\lambda p. \ln (\text{real} p) / (p \text{pour} x - 1))\) abs-summable-on \(\{p. \text{prime} p\}\) \((\text{is} \ ?\text{th}1)\)

and \(\text{deriv zeta} (\text{of-real} x) / \text{zeta} (\text{of-real} x) = -\text{of-real} (\sum p | \text{prime} p. \ln (\text{real} p) / (p \text{pour} x - 1))\) \((\text{is} \ ?\text{th}2)\)

proof –

have \((\lambda p. \Re (\text{of-real} (\ln (\text{real} p)) / (\text{of-nat} p \text{pour of-real} x - 1)))\)

abs-summable-on \(\{p. \text{prime} p\}\) using assms

by (intro abs-summable-\(\Re\) eval-fds-logderiv-zeta) auto

also have \(?\text{this} \iff \?\text{th}1\)

by (intro abs-summable-on-cong) (auto simp: powr-Reals-eq)

finally show \(?\text{th}1\).

show \(?\text{th}2\) using assms

by (subst eval-fds-logderiv-zeta) (auto simp: infsetsum-of-real [symmetric] powr-Reals-eq)

qed

lemma

fixes a b c d :: real

assumes ab: \(d * a + b \geq 1\) \(\text{and} c: c < -1\) \(\text{and} d: d > 0\)

defines \(C \equiv - ((\ln (d * a + b) - 1) / (c + 1) ) * (d * a + b) \text{pour} (c + 1) / (d * (c + 1)))\)

shows set-integrable-ln-pour-at-top:

\((\lambda x. (\ln (d * x + b) * ((d * x + b) \text{pour} c)))\) absolutely-integrable-on \(\{a..\}\) \((\text{is} \ ?\text{th}1)\)

and set-lebesgue-integrable-ln-pour-at-top:

\(\{f x\in\{a..\}. (\ln (d * x + b) * ((d * x + b) \text{pour} c)) \partial\text{borel}\} = C\) \((\text{is} \ ?\text{th}2)\)

and \(\text{ln-pour-has-integral-at-top}\)
\((\lambda x. \ln (d \ast x + b) \ast (d \ast x + b) \ \text{powr} \ c) \ \text{has-integral} \ C) \ \{a<..\} \ (\text{is ?th3})\)

**proof** –

**define** \(f\) where \(f = (\lambda x. \ln (d \ast x + b) \ast (d \ast x + b) \ \text{powr} \ c)\)

**define** \(F\) where \(F = (\lambda x. (\ln (d \ast x + b) - 1 / (c + 1)) \ast (d \ast x + b) \ \text{powr} \ (c + 1)) / (d \ast (c + 1)))\)

**have** \(*\): \((F \ \text{has-field-derivative} \ f \ x) \ (at \ x) \ \text{isCont} \ f \ f \ x \ \geq 0 \ \text{if} \ x > a \ \text{for} \ x\)

**proof** –

**have** \(1 \leq d \ast a + b\ \text{by} \ \text{fact}\)

**also have** \(\ldots < d \ast x + b\ \text{using that assms}\)

**finally have** \(gt-1: d \ast x + b > 1\).

**show** \((F \ \text{has-field-derivative} \ f \ x) \ (at \ x) \ \text{isCont} \ f \ x \ \text{using} \ ab \ c \ d \ gt-1\)

**by** \((\text{auto simp: F-def f-def divide-simps intro: derivative-eq-intros continuous-intros})\)

**finally have** \(gt-1: d \ast x + b > 1\).

**show** \((F \ \text{has-field-derivative} \ f \ x) \ (at \ x) \ \text{isCont} \ f \ x\)

**using** \(\text{ab c d gt-1} \ \text{by} \ (\text{auto simp: algebra-simps powr-add})\)

**show** \(f \ x \ \geq 0\)

**using** \(gt-1\) \ \text{by} \ ((\text{auto simp: f-def})\)

**qed**

**have** \(\text{limits:} ((F \circ \text{real-of-ereal} ) \longrightarrow F \ a) \ (\text{at-right} \ (\text{ereal} \ a))\)

**using** \(ab \ c \ \text{limits}\)

**thus** \(2: f \ \text{absolutely-integrable-on} \ (a<..)\)

**by** \((\text{auto simp: set-integrable-def integrable-completion})\)

**have** \((\text{LBINT} x=\text{ereal} a..\infty. f x) = 0 \ - \ F a \ \text{using ab c limits}\)

**by** \((\text{auto simp: interval-integral-FTC-nonneg})\)

**thus** \(3: \ ?\text{th2}\)

**by** \((\text{simp add: interval-to-infinity-eq F-def f-def C-def})\)

**show** \(\ ?\text{th3}\)

**using** \(\text{set-borel-integral-eq-integral[OF 1] 3 by} \ (\text{simp add: has-integral-iff f-def C-def})\)

**qed**

**lemma** \(\text{ln-fact-conv-sum-upto}: \ln (\text{fact} \ n) = \text{sum-upto} \ \ln \ n\)

**by** \((\text{induction} \ n) \ (\text{auto simp: sum-upto-plus1 add.commute[of 1] ln-mult})\)

**lemma** \(\text{sum-upto-ln-conv-ln-fact}: \text{sum-upto} \ \ln \ x = \ln (\text{fact} \ (\text{nat} \ \lfloor x \rfloor))\)

**by** \((\text{simp add: ln-fact-conv-sum-upto sum-upto-altdef})\)

**lemma** \(\text{real-of-nat-div}: \text{real} (\text{a div b}) = \text{real-of-int} \ [\text{real} \ a / \text{real} \ b]\)

**by** \((\text{subst floor-divide-of-nat-eq})\)

**lemma** \(\text{integral-subset-negligible}:\)

**fixes** \(f :: 'a :: \text{euclidean-space} \Rightarrow 'b :: \text{banach}\)

**assumes** \(S \subseteq T \ \text{negligible} \ (T - S)\)

**shows** \((\text{integral} \ S \ f = \text{integral} \ T \ f)\)

**proof** –

45
have \( \int T f = \int (\lambda x. \text{if } x \in S \text{ then } f x \text{ else } 0) \)
by \(\text{rule integral-spike[of } T - S\)] (use assms in auto)
also have \(\ldots = \int (S \cap T) f\)
by \(\text{subst integral-restrict-Int} \) auto
also have \(S \cap T = S\) using assms by auto
finally show \(?thesis ..\)
qed

lemma integrable-on-cong [cong]:
assumes \(\forall x. x \in A \implies f x = g x\)
shows \(f\) integrable-on \(A\) \(\iff\) \(g\) integrable-on \(B\)
using has-integral-cong [of \(A\) \(f\) \(g\) \(\text{OF assms(1)}\)] \(\text{assms(2)}\)
by \(\text{auto simp:integrable-on-def}\)

lemma measurable-sum-upto [measurable]:
fixes \(f :: 'a \Rightarrow \text{nat} \Rightarrow \text{real}\)
assumes \([\text{measurable}]: \forall y. (\lambda t. f t y) \in M \rightarrow M\) borel
assumes \([\text{measurable}]: x \in M \rightarrow M\) borel
shows \((\lambda t. \text{sum-upto } (f t) (x t)) \in M \rightarrow M\) borel
proof
have \(\text{meas} : (\lambda t. \text{set-lebesgue-integral } lborel \{y. y \geq 0 \land y - \text{real } |x t| \leq 0\}) (\lambda y. f t (\text{nat }|y|)) \in M \rightarrow M\) borel
(is \(?f \in -\) unfolding set-lebesgue-integral-def)
by measurable
also have \(?f = (\lambda t. \text{sum-upto } (f t) (x t))\)
proof
fix \(t :: 'a\)
show \(?f t = \text{sum-upto } (f t) (x t)\)
proof \(\text{(cases } x t < 1)\)
case True
hence \(\{y. y \geq 0 \land y - \text{real } |x t| \leq 0\} = \{0\}\) by auto
thus \(?thesis \) using True
by \(\text{simp add: set-integral-at-point sum-upto-altdef}\)
next
case False
define \(n\) where \(n = \text{nat } |x t|\)
from False have \(n > 0\) by \(\text{auto simp:n-def}\)
have \(\ast: ((\lambda x. f t (\text{nat } |x|)) \text{ has-integral sum } (f t) \{0<..n\}) \text{ \{real }0\ldots\text{real }n\}\)
using \(\langle n > 0\rangle\) by \(\text{intro nat-sum-has-integral-ceiling} \) auto

have \(\ast\ast: (\lambda x. f t (\text{nat } |x|)) \text{ absolutely-integrable-on } \{\text{real }0\ldots\text{real }n\}\)
proof \(\text{(rule absolutely-integrable-absolutely-integrable-ubound)}\)
show \(\lambda t. \text{MAX } n \in \{0..n\}. |f t n| \text{ absolutely-integrable-on } \{\text{real }0\ldots\text{real }n\}\)
using \(\langle n > 0\rangle\) by \(\text{subst absolutely-integrable-on-iff-nonneg}\)
(auto simp: Max-ge-iff intro: \text{exI[of } f t 0\})
show \(\lambda x. f t (\text{nat } |x|)) \text{ integrable-on } \{\text{real }0\ldots\text{real }n\}\)
using \(\ast\) by \(\text{simp add: has-integral-iff}\)
next
2 Ingham’s Tauberian Theorem

theory Newman-Ingham-Tauberian

imports
   HOL-Real-Asymp.Real-Asymp
   Prime-Number-Theorem-Library

begin

In his proof of the Prime Number Theorem, Newman [6] uses a Tauberian theorem that was first proven by Ingham. Newman gives a nice and straightforward proof of this theorem based on contour integration. This section will be concerned with proving this theorem.

This Tauberian theorem is probably the part of the Newman’s proof of the Prime Number Theorem where most of the “heavy lifting” is done. Its purpose is to extend the summability of a Dirichlet series with bounded coefficients from the region $\Re(s) > 1$ to $\Re(s) \geq 1$.

In order to show it, we first require a number of auxiliary bounding lemmas.

lemma newman-ingham-aux1:
   fixes $R :: real$ and $z :: complex$
   assumes $R : R > 0$ and $z : \text{norm } z = R$
   shows $\text{norm } (1 / z + z / R^2) = 2 * |\text{Re } z| / R^2$

proof –
   from $z$ and $R$ have [simp]: $z \neq 0$ by auto
   have $1 / z + z / R^2 = (R^2 + z^2) * (1 / R^2 / z)$ using $R$
Given a function that is analytic on some vertical line segment, we can find a rectangle around that line segment on which the function is also analytic.

**Lemma analytic-on-axis-extend:**

```
by (simp add: field-simps power2-eq-square)
also have norm ... = norm (R^2 + z^2) / R 3
  by (simp add: numeral-3-eq-3 z norm-divide norm-mul power2-eq-square)
also have R^2 + z^2 = z * (z + cnj z) using complex-norm-square[of z]
  by (simp add: z power2-eq-square algebra-simps)
also have norm ... = 2 * |Re z| * R
  by (subst complex-add-cnj) (simp-all add: z norm-mult)
also have ... / R ^ 3 = 2 * |Re z| / R^2
  using R by (simp add: field-simps numeral-3-eq-3 power2-eq-square)
finally show ?thesis .
```

**QED**

**Lemma newman-ingham-aux2:**

```
fixes m :: nat and w z :: complex
assumes 1 \leq m 1 \leq Re w 0 < Re z and f: \forall n. 1 \leq n \Rightarrow norm (f n) \leq C
shows norm (\sum n=1..m. f n / n powr (w - z)) \leq C * (m powr Re z) * (1 / m + 1 / Re z)
```

**Proof**

```
proof
  have [simp]: C \geq 0 by (rule order.trans[OF f[of 1]]) auto
  have norm (\sum n=1..m. f n / n powr (w - z)) \leq (\sum n=1..m. C / n powr (1 - Re z))
    by (rule sum-norm-le)
      (insert assms, auto simp: norm-divide norm-powr-real-powr intro!: frac-le assms powr-mono)
  also have ... = C * (\sum n=1..m. n powr (Re z - 1))
    by (subst sum-distrib-left) (simp-all add: powr-diff)
  also have ... \leq C * (m powr Re z * (1 / Re z + 1 / m))
    using zeta-partial-sum-le[of Re z m] assms by (intro mult-left-mono) auto
finally show ?thesis by (simp add: mult-ac add-ac)
```

**QED**

**Lemma hurwitz-zeta-real-bound-aux:**

```
fixes a x :: real
assumes ax: a > 0 x > 1
shows (\sum i. (a + real (Suc i)) powr (- x)) \leq a powr (1 - x) / (x - 1)
```

**Proof**

```
proof (rule decreasing-sum-le-integral, goal-cases)
  have ((\lambda t. (a + t) powr -x) has-integral -(a powr (- x + 1)) / (-x + 1))
    (interior {0..})
      using powr-has-integral-at-top[of 0 a -x] using ax by (simp add: interior-real-semiline)
  also have -(a powr (- x + 1)) / (- x + 1) = a powr (1 - x) / (x - 1)
    using ax by (simp add: field-simps)
  finally show ((\lambda t. (a + t) powr -x) has-integral a powr (1 - x) / (x - 1))
    \{0..\}
      by (simp add: has-integral-interior)
```

**QED**

```
(insert ax, auto intro!: powr-mono2)
```

Given a function that is analytic on some vertical line segment, we can find a rectangle around that line segment on which the function is also analytic.

**Lemma analytic-on-axis-extend:**
fixes $y_1$ $y_2$ $x$ :: real
defines $S \equiv \{ z. \operatorname{Re} z = x \land \operatorname{Im} z \in \{ y_1 .. y_2 \} \}$
assumes $y_1 \leq y_2$
assumes $f$ analytic-on $S$
obtaines $x_1$ $x_2$ :: real where $x_1 < x x_2 > x$ analytic-on cbox $(\text{Complex } x_1 y_1) (\text{Complex } x_2 y_2)$

\textbf{proof} –
\begin{align*}
define C where C = \{ box a b | a b z. f \text{ analytic-on box } a b \land z \in \text{box } a b \land z \in S \} \\
have $S = \text{cbox } (\text{Complex } x y_1) (\text{Complex } x y_2)$ \\
by (auto simp: $S$-def in-cbox-complex-iff)
also have compact … by simp \\
finally have 1: compact $S$ .
\end{align*}

have 2: $S \subseteq \bigcup C$
\begin{align*}
\text{proof} (\text{intro subsetI}) \\
\text{fix } z \text{ assume } z \in S \\
\text{from } f \text{ analytic-on } S; \text{ and this obtain } a b \text{ where } z \in \text{box } a b f \text{ analytic-on box } a b \\
by (\text{blast elim: analytic-onE-box}) \\
\text{with } (z \in S) \text{ show } z \in \bigcup C \text{ unfolding } C$-def by blast
\end{align*}
\begin{proof}
have $3$: open $X$ if $X \in C$ for $X$ using that by (auto simp: $C$-def) \\
from compactE[OF 1 2 3] obtain $T$ where $T: T \subseteq C$ finite $T S \subseteq \bigcup T$ \\
by blast
\begin{align*}
define x_1 \text{ where } x_1 = \text{Max } (\text{insert } (x - 1) (\lambda X. x + (\text{Inf } (\operatorname{Re} X) - x) / 2) \cdot T)) \\
define x_2 \text{ where } x_2 = \text{Min } (\text{insert } (x + 1) (\lambda X. x + (\text{Sup } (\operatorname{Re} X) - x) / 2) \cdot T))
\end{align*}

have $\ast$: $x + (\text{Inf } (\operatorname{Re} X) - x) / 2 < x \land x + (\text{Sup } (\operatorname{Re} X) - x) / 2 > x$ if $X \in T$ for $X$
\begin{proof}
\text{from that and } T \text{ obtain } a b s \text{ where } [\text{simp}]: X = box a b \text{ and } s: s \in box a b s \in S \\
by (force simp: $C$-def) \\
hence le: Re $a < Re b \Im a < \Im b$ by (auto simp: in-box-complex-iff)
\text{show } \ast\text{thesis using } le s \\
\text{unfolding } X = box a b \text{ Re-image-box}[OF le] \text{ Im-image-box}[OF le] \\
by (auto simp: $S$-def in-box-complex-iff)
\end{proof}
\begin{proof}
from $\ast$ $T$ have $x_1 < x$ unfolding $x_1$-def by (subst Max-less-iff) auto \\
from $\ast$ $T$ have $x_2 > x$ unfolding $x_2$-def by (subst Min-gr-iff) auto
\end{proof}
have $f$ analytic-on $(\bigcup T)$ \\
using $T$ by (subst analytic-on-Union) (auto simp: $C$-def) \\
moreover have $z \in \bigcup T$ if $z \in \text{cbox } (\text{Complex } x_1 y_1) (\text{Complex } x_2 y_2)$ for $z$
proof 
from that have Complex x (Im z) ∈ S
by (auto simp: in-box-complex-iff S-def)
with T obtain X where X: X ∈ T Complex x (Im z) ∈ X
by auto
with T obtain a b where [simp]: X = box a b by (auto simp: C-def)
from X have le: Re a < Re b Im a < Im b by (auto simp: in-box-complex-iff)

from that have Re z ≤ x^2 by (simp add: in-box-complex-iff)
also have ... ≤ x + (Sup (Re ' X) - x) / 2
  unfolding x^2-def by (rule Min.coboundedI)(use T X in auto)
also have ... = (x + Re b) / 2
  using le unfolding (X = box a b) Re-image-box[OF le] by (simp add: field-simps)
also have ... < (Re b + Re b) / 2
  using X by (intro divide-strict-right-mono add-strict-right-mono)
  (auto simp: in-box-complex-iff)
also have ... = Re b by simp
finally have [simp]: Re z < Re b.

have Re a = (Re a + Re a) / 2 by simp
also have ... < (x + Re a) / 2
  using X by (intro divide-strict-right-mono add-strict-right-mono)
  (auto simp: in-box-complex-iff)
also have ... = x + (Inf (Re ' X) - x) / 2
  using le unfolding (X = box a b) Re-image-box[OF le] by (simp add: field-simps)
also have ... ≤ x1 unfolding x1-def by (rule Max.coboundedI)(use T X in auto)
also have ... ≤ Re z using that by (simp add: in-box-complex-iff)
finally have [simp]: Re z > Re a.

from X have z ∈ X by (simp add: in-box-complex-iff)
with T X show ?thesis by blast
qed

hence box (Complex x1 y1) (Complex x2 y2) ⊆ ∪ T by blast
ultimately have f analytic-on cbox (Complex x1 y1) (Complex x2 y2)
  by (rule analytic-on-subset)

with (x1 < x) and (x2 > x) and that[of x1 x2] show ?thesis by blast
qed

We will now prove the theorem. The precise setting is this: Consider a Dirichlet series \( F(s) = \sum a_n n^{-s} \) with bounded coefficients. Clearly, this converges to an analytic function \( f(s) \) on \( \{ s \mid \Re(s) > 1 \} \).
If \( f(s) \) is analytic on the larger set \( \{ s \mid \Re(s) \geq 1 \} \), \( F \) converges to \( f(s) \) for all \( \Re(s) \geq 1 \).
The proof follows Newman’s argument very closely, but some of the precise
bounds we use are a bit different from his. Also, like Harrison, we choose a combination of a semicircle and a rectangle as our contour, whereas Newman uses a circle with a vertical cut-off. The result of the Residue theorem is the same in both cases, but the bounding of the contributions of the different parts is somewhat different.

The reason why we picked Harrison’s contour over Newman’s is because we could not understand how his bounding of the different contributions fits to his contour, and it seems likely that this is also the reason why Harrison altered the contour in the first place.

**Theorem** Newman-Ingham:

- **Fixes** \( F :: \text{complex} \) and \( f :: \text{complex} \Rightarrow \text{complex} \)
- **Assumes** coeff-bound: \( \text{fds-nth } F \in O(\lambda , 1) \)
- **Assumes** f-analytic: \( f \text{ analytic-on } \{ s. \text{Re } s \geq 1 \} \)
- **Assumes** F-conv-f: \( \forall s. \text{Re } s > 1 \Rightarrow \text{eval-fds } F s = f s \)
- **Assumes** conv-absissa: \( \text{conv-absissa } F \leq 1 \)
- **Show** \( \text{fds-converges } F w \) and \( \text{eval-fds } F w = f w \)

**Proof** —

- We get a bound on our coefficients and call it \( C \).
- **Obtain** \( C \) where \( C \geq 1 \) \( \land n. \text{norm } (\text{fds-nth } F n) \leq C \)

- **Using** natfun-bigo-1E[OF coeff-bound, where \( lb = 1 \)] by blast

**Write** contour-integral \( \oint \)

— We show convergence directly by showing that the difference between the partial sums and the limit vanishes.

- **Have** \( \lambda N. \text{eval-fds } (\text{fds-truncate } N F) w \longrightarrow f w \)
- **Unfolding** tendssto-iff dist-norm norm-minus-commute[of eval-fds F s for F s]

**Proof** —

- **Fix** \( \varepsilon :: \text{real} \) **Assume** \( \varepsilon > 0 \)

— We choose an integration radius that is big enough for the error to be sufficiently small.

- **Define** \( R \) where \( R = \max 1 (3 * C / \varepsilon) \)
- **Have** \( R \geq 3 * C / \varepsilon \) \( \Rightarrow R \geq 1 \) by (auto simp: R-def)

— Next, we extend the analyticity of \( f (w + z) \) to the left of the complex plane within a thin rectangle that is at least as high as the circle.

- **Obtain** \( l \) where \( l > 0 \)

\( \lambda z. f (w + z) \) analytic-on \( \{ s. \text{Re } s < 0 \lor \text{Im } s \in \{ -R - 1 .. < R + 1 \} \land \text{Re } s > -l \} \)

**Proof** —

- **Have** f-analytic': \( \lambda z. f (w + z) \) analytic-on \( \{ s. \text{Re } s \geq 0 \} \)
  - by (rule analytic-on-compose-gen[OF - f-analytic, unfolded o-def])
  - (insert w, auto intro: analytic-intros)
- **Hence** \( \lambda z. f (w + z) \) analytic-on \( \{ s. \text{Re } s = 0 \land \text{Im } s \in \{ -R - 1 .. R + 1 \} \} \)
  - by (rule analytic-on-subset) auto

- **From** analytic-on-axis-extend[OF - this] **Obtain** \( x1 \) \( x2 \) **Where** \( x12: \)

\( x1 < 0 \land x2 > 0 \) \( \lambda z. f (w + z) \) analytic-onobox (Complex x1 \( \{ -R - 1 \})

(Complex x2 \( \{ R + 1 \}))
using \( R \geq 1 \) by auto
from this(5) have \((\lambda z. f (w + z)) \text{ analytic-on } \{ s. \ Re s \in \{x1..0\} \wedge \text{Im } s \in \{-R-1..R+1\}\}\)
by (rule analytic-on-subset) (insert \( x12 \), auto simp: in-cbox-complex-iff)
with \( f\)-analytic' have \((\lambda z. f (w + z)) \text{ analytic-on } \)
\(\{ s. \ Re s \geq 0 \} \cup \{ s. \ Re s \in \{x1..0\} \wedge \text{Im } s \in \{-R-1..R+1\}\}\)
by (subst analytic-on-Un) auto
hence \((\lambda z. f (w + z)) \text{ analytic-on } \{ s. \ Re s > 0 \lor \text{Im } s \in \{-R-1..<..R+1\}\} \wedge \text{Re } s > x1\)
by (rule analytic-on-subset) auto
with \((x1 < 0) \text{ and that[of } -x1]\) show \(?thesis\) by auto
qed

— The function \( f (w + z) \) is now analytic on the open box \((-l; R+1) + i(-R + 1; R+1)\). We call this region \( X\).

**define** \( X \) **where** \( X = \text{box (Complex (-l) (-R-1)) (Complex (R+1) (R+1))}\)
**have** [simp, intro]: \( \text{open } X \text{ convex } X \) by (simp-all add: X-def open-box)
**from** \( R \lor \text{have } [simp]: 0 \in X \) **by** (auto simp: X-def in-box-complex-iff)
**have** analytic: \((\lambda z. f (w + z)) \text{ analytic-on } X\)
by (rule analytic-on-subset[OF l(2)]) (auto simp: X-def in-box-complex-iff)

**note** \( f\)-analytic' [analytic-intros] = analytic-on-compose-gen[OF - analytic, unfolded o-def]

**note** \( f\)-holo [holomorphic-intros] =
holomorphic-on-compose-gen[OF - analytic-imp-holomorphic[OF analytic], unfolded o-def]

**note** \( f\)-cont [continuous-intros] = continuous-on-compose2[OF holomorphic-on-imp-continuous-on[OF analytic-imp-holomorphic[OF analytic]]]

— We now pick a smaller closed box \( X' \) inside the big open box \( X\). This is because we need a compact set for the next step. Our integration path still lies entirely within \( X'\), and since \( X'\) is compact, \( f (w + z) \) is bounded on it, so we obtain such a bound and call it \( M\).

**define** \( \delta \) **where** \( \delta = \text{min (1/2) (l/2)}\)
**from** \( l \) **have** \( \delta \geq 0 \text{ and } 1/2 \delta < l \) **by** (auto simp: \( \delta\)-def)
**define** \( X' \) **where** \( X' = \text{box (Complex (-\delta) (-R)) (Complex R R)}\)
**have** \( X' \subseteq X \) **unfolding** \( X'\)-def **X-def using** l(1) \( R \delta\)
by (intro subset-box-imp) (auto simp: Basis-complex-def)
**have** [intro]: \( \text{compact } X' \) **by** (simp add: X'-def)
**moreover have** continuous-on \( X' (\lambda z. f (w + z))\)
using \( w \) \( X' \subseteq X\) **by** (auto intro!: continuous-intros)
ultimately obtain \( M \) **where** \( M: M \geq 0 \land z \in X' \Rightarrow \text{norm } (f (w + z)) \leq M\)
using continuous-on-compact-bound by blast

— Our objective is now to show that the difference between the \( N\)-th partial sum and the limit is below a certain bound (depending on \( N\)) which tends to \( 0 \) for \( N \to \infty\). We use the following bound:
See image for text
using δ by (auto simp: B1-def B3-def closed-segment-same-Im closed-segment-eq-real-iel)

have image-A: path-image A = {s. Re s ≥ 0 ∧ norm s = R}

unfolding A-def using R by (subst path-image-semicolon-Re-ge) auto

also have z ∈ ... → z ∈ X' − {∅} for z

using complex-Rem-le-cmod[of z] abs-Im-le-cmod[of z] δ R

by (auto simp: X'-def in-cbox-complex-iff)

hence {s. Re s ≥ 0 ∧ norm s = R} ⊆ X' − {∅} by auto

finally have path-image B2 ⊆ X' − {∅} path-image A ⊆ X' − {∅} using δ > 0;

by (auto simp: X'-def in-cbox-complex-iff image-B2 image-B1 image-B3)

note path-images = this (X' ⊆ X)

— Γ is a simple path, which, combined with its simple geometric shape, makes reasoning about its winding numbers trivial.

from R have simple-path A unfolding A-def

by (subst simple-path-part-circlepath) auto

have simple-path Γ unfolding Γ-def

proof (intro simple-path-join-loop subsetI arc-join, goal-cases)

fix z assume z: z ∈ path-image A ∩ path-image (B1 +++ B2 +++ B3)

with image-A have Re z ≥ 0 norm z = R by auto

with z R & show z ∈ {Pathstart A, Pathstart (B1 +++ B2 +++ B3)}

by (auto simp: path-image-join-image-B1 image-B2 image-B3 complex-eq-iff)

qed (insert R, auto simp: image-B1 image-B3 path-image-join image-B2 complex-eq-iff)

— We define the integrands in the same fashion as Newman:

define g where g = (λz::complex. f (w + z) * N powr z * (1 / z + z / R\(^2\))

define S where S = eval-fds (fds-truncate N F)

define g-S where g-S = (λz::complex. S (w + z) * N powr z * (1 / z + z / R\(^2\))

define rem where rem = (λz::complex. f z − S z)

define g-rem where g-rem = (λz::complex. rem (w + z) * N powr z * (1 / z + z / R\(^2\)))

have g-holo: g holomorphic-on X − {∅} unfolding g-def

by (auto intro!: holomorphic-intros analytic-imp-holomorphic[of analytic])

have rem-altdef: rem z = eval-fds (fds-remainder N F) z if Re z > 1 for z

proof —

from assms and that have f z = eval-fds F z by auto

also have F = fds-truncate N F + fds-remainder N F

by (rule fds-truncate-plus-remainder [symmetric])

also from that have eval fds ... z = S z + eval fds (fds-remainder N F)

z unfolding S-def

by (subst eval-fds-add) (auto intro: fds-converges[of le-lesstrans[of converge-absisse]])

finally show thesis by (simp add: rem-def)

qed
We now come to the first application of the residue theorem along the path $\Gamma$:

$$\oint_{\Gamma} g = 2 \pi i \cdot \text{winding-number } \Gamma \cdot 0 \cdot \text{residue } g \cdot 0$$

**proof** (subst Residue-theorem)

show $g$ holomorphic-on $X - \{0\}$ by fact

show path-image $\Gamma \subseteq X - \{0\}$ using path-images

by (auto simp: $\Gamma$-def path-image-join)

thus $\forall z. z \notin X \implies \text{winding-number } \Gamma \cdot z = 0$

by (auto intro!: simply-connected-imp-winding-number-zero[of $X$] convex-imp-simply-connected)

qed (insert path-images, auto intro: convex-connected)

also have $\text{winding-number } \Gamma \cdot 0 = 1$

**proof** (rule simple-closed-path-winding-number-pos)

from $R \cdot \delta$ have $\forall g \in \{A, B1, B2, B3\}$. $\text{Re}(\text{winding-number } g \cdot 0) > 0$

unfolding $A$-def $B1$-def $B2$-def $B3$-def

by (auto intro!: winding-number-linepath-pos_lt winding-number-part-circlepath-pos_less)

hence valid-path $\Gamma \land 0 \notin \text{path-image } \Gamma \land \text{Re}(\text{winding-number } \Gamma \cdot 0) > 0$

unfolding $\Gamma$-def using path-images (1-4) by (intro winding-number-join-pos-combined)

auto

thus $\text{Re}(\text{winding-number } \Gamma \cdot 0) > 0$ by simp

qed (insert path-images, auto intro: convex-connected)

also have $\text{residue } g \cdot 0 = f \cdot w$

**proof**

- have $g = (\lambda z::\text{complex}. f (w + z) \cdot N \cdot \text{powr } z \cdot (1 + z^2 / R^2) / z)$

  by (auto simp: $g$-def divide-simps fun-eq_iff power2_eq_square simp del: div_mult_self3 div_mult_self4 div_mult_self2 div_mult_self1)

  moreover from $N$ have $\text{residue } g \cdot 0 = f \cdot w$

  by (subst residue-simpleꦽ[of $X$])

  (auto intro!: holomorphic-intros analytic-imp-holomorphic[OF analytic])

ultimately show $?\text{thesis}$ by (simp only:)

qed

finally have $2 \cdot \pi i \cdot f \cdot w = \oint_{\Gamma} g$ by simp

also have $\ldots = \oint [A] g + \oint [B2] g + \oint [B1] g + \oint [B3] g$ unfolding $\Gamma$-def

by (subst contour-integral-join, (insert path-images, auto intro!: contour-integral-join contour-integrable-holomorphic-simple $g$-holy[4]+)

(simp-all add: add-ac)

finally have integral1: $2 \cdot \pi i \cdot f \cdot w = \oint [A] g + \oint [B2] g + \oint [B1] g + \oint [B3] g$.

Next, we apply the residue theorem along a circle of radius $R$ to another integrand that is related to the partial sum:

**have** $\oint \text{circlepath } 0 \cdot R \cdot g \cdot S = 2 \cdot \pi i \cdot \text{residue } g \cdot S \cdot 0$

**proof** (subst Residue-theorem)

show $g \cdot S$ holomorphic-on $\text{UNIV} - \{0\}$

by (auto simp: $g$-S-def $S$-def intro!: holomorphic-intros)

qed (insert $R$, auto simp: winding-number-circlepath-centre)

also have $\text{residue } g \cdot S \cdot 0 = S \cdot w$

**proof**
have \( g \cdot S = (\lambda z :: \text{complex}. \ S \ (w + z) \ast N \ \text{powr} \ z \ast (1 + z^2 / R^2) / z) \)
\begin{align*}
& \text{by (auto simp: g-S-def divide-simps fun-eq-iff power2-eq-square simp del: div-mu}\ldots) \\
& \text{moreover from } N \text{ have residue ... } 0 = S \ w \\
& \text{by (subst residue-simple[of } X])} \\
& \text{(auto intro!: holomorphic-intros simp: S-def)} \\
& \text{ultimately show } \text{?thesis by (simp only:)} \\
& \text{qed} \\
& \text{finally have } 2 \ast \pi \ast i \ast S \ w = \oint [\text{circlepath } 0 \ R] \ g \cdot S \ ..
\end{align*}

— We split this integral into integrals along two semicircles in the left and right half-plane, respectively:
\begin{enumerate}
\item \text{also have ... } = \oint [\text{part-circlepath } 0 \ R \ (-\pi/2) \ (3\pi/2)] \ g \cdot S \\
\item \text{proof (rule Cauchy-theorem-homotopic-loops)} \\
\item \text{show homotopic-loops } (-\{0\}) \ (\text{circlepath } 0 \ R) \\
\item \text{(part-circlepath } 0 \ R \ (-\pi/2) \ (3\pi/2)) \text{ unfolding circlepath-def using } R \\
\item \text{by (intro homotopic-loops-part-circlepath[where } k = 1]) \text{ auto} \\
\item \text{proof (rule Cauchy-theorem-homotopic-paths)} \\
\item \text{have } \ast : -A = \text{part-circlepath } 0 \ R \ (\pi/2) \ (3\pi/2) \text{ unfolding A-def} \\
\item \text{by (intro part-circlepath-mirror[where } k = 0]) \text{ auto} \\
\item \text{from } R \text{ show homotopic-paths } (-\{0\}) \ (\text{part-circlepath } 0 \ R \ (-\pi/2) \\
\item \text{(3\pi/2)) (A +++ -A)} \\
\item \text{unfolding \ast \ unfolding A-def} \\
\item \text{by (intro homotopic-paths-part-circlepath) (auto dest!: in-path-image-part-circlepath) \\
\item \text{proof (auto simp: g-S-def S-def intro!: holomorphic-intros)} \\
\item \text{also have ... } = \oint [\lambda x. g \cdot S - \oint [\lambda A. g \cdot S] (\lambda x. g \cdot S (-x)) \\
\item \text{by (simp add: A-def contour-integral-mirror contour-integral-ineq) \\
\item \text{finally have integral2: } 2 \ast \pi \ast i \ast S \ w = \oint [\lambda A. g \cdot S - \oint [\lambda A. g \cdot S] (\lambda x. g \cdot S (-x)) \\
\item \text{by simp}
\end{enumerate}

— Next, we show a small bounding lemma that we will need for the final estimate:
\begin{enumerate}
\item \text{have circle-bound: norm } (1 / z + z / R^2) \leq 2 / R \text{ if } \text{simp: norm } z = R \\
\end{enumerate}

for \( z :: \text{complex} \)
\begin{enumerate}
\item \text{proof ...} \\
\item \text{have norm } (1 / z + z / R^2) \leq 1 / R + 1 / R \\
\item \text{by (intro order.trans[OF norm-triangle-ineq] add-mono) \\
\item \text{(insert } R, \text{ simp-all add: norm-divide norm-mult power2-eq-square)} \\
\item \text{thus ?thesis by simp}
\end{enumerate}

\text{qed}
— The next bound differs somewhat from Newman’s, but it works just as well. Its purpose is to bound the contribution of the two short horizontal line segments. 

**have B12-bound:** norm (integral \{- \delta..0\} (λx. \, g (x + R' * i))) ≤ 3 * M / \( R / \ln N \)

(is \( |I| ≤ \) if \( |R'| = R \) for \( R' \))

**proof**

- **have** \( |I| ≤ \) integral \{- \delta..0\} (λx. \, 3 * M / \( R * N \) powr x)
- **proof** (rule integral-norm-bound-integral)
  - fix \( x \) assume \( x: x ∈ \{-\delta..0\} \)
  - define \( z \) where \( z = x + i * R' \)
  - from \( R \) that **have** [simp]: \( z = 0 \) \( Re \, z = x \) \( Im \, z = R' \)
  - by (auto simp: z-def complex-eq-iff)
- from \( x R \) that **have** \( z ∈ X' \) by (auto simp: z-def X′-def in-cbox-complex-iff)
- from \( x R \) that **have** norm \( z ≤ \delta + R \)
- by (intro order.trans[OF cmod-le add-mono]) auto

  hence norm \( (1 / z + z / R^2) ≤ 1 / (R + (\delta / (R + 1)) / R \)
  - using \( R \) that abs-Im-le-cmod[of \( z \) ]
  - by (intro order.trans[OF OF norm-triangle-ineq add-mono])
    (auto simp: norm-divide norm-mult power2-eq-square field-simps )

  also have \( \delta / R ≤ 1 \) using \( \delta \) \( R \) by auto

  finally have norm \( (1 / z + z / R^2) ≤ 3 / R \)
  - using \( R \) by (simp add: divide-right-mono)

  hence norm \( (g z) ≤ M * N \) powr \( x * (3 / R) \)
  - unfolding g-def norm-mult using \( M ≥ 0 \) \( \langle z ∈ X' \rangle \)
  - by (intro mult-mono mult-nonneg-nonneg M) (auto simp: norm-powr-real-powr)

  thus norm \( (g (x + R' * i)) ≤ 3 * M / (R * N) \) powr \( x \) by (simp add: mult-ac z-def)

  **qed** (insert \( N \) \( R \) \( l \) that \( \delta \), auto intro!: integrable-continuous-real continuous-intros
    simp: g-def X-def complex-eq-iff in-box-complex-iff)

  also have \( \ldots = 3 * M / R * \) integral \{-\delta..0\} (λx. \, N powr x) by simp

  also have \( ((λx. \, N \) powr x) has-integral \( (N \) powr 0 / \ln \( N - N \) powr \(-\delta) \)
    / \ln N) \{-\delta..0\} \)

  using \( \delta \) \( N \)
  - by (intro fundamental-theorem-of-calculus)
    (auto simp: has-field-derivative-iff has-vector-derivative [symmetric]
    powr-def
  
    intro!: derivative-eq-intros)

  hence integral \{-\delta..0\} (λx. \, N powr x) = 1 / \ln \( \) (real \( N \)) = \ln \( \) (real \( N \))

  using \( N \) by (simp add: has-integral-iff)

  also have \( \ldots ≤ 1 / \ln \) (real \( N \)) using \( N \) by simp

  finally show ?thesis using \( M \) \( R \) by (simp add: mult-left-mono divide-right-mono)

  **qed**
\[ \oint \] unfolding ring-distribrs integral1 integral2 by (simp add: algebra-simps)
also have \[ \oint [A] g - \oint [A] g-S = \oint [A] (\lambda x. g x - g-S x) \] using path-images
by (intro contour-integral-diff [symmetric])
(auto intro: contour-integrable-holomorphic-simple[of - X - {0}])

holomorphic-intrs
simp: g-S-def g-holo S-def
also have \[ \ldots = \oint [A] g-rem \]
by (simp add: g-rem-def g-def algebra-simps rem-def)
finally have \[ 2 \ast \pi \ast i \ast (f w - S w) = \]
\[ \oint [A] g-rem + \oint [A] (\lambda x. g-S (- x)) + \oint [B1] g + \oint [B3] g + \oint [B2] g . \]

— We now bound each of these integrals individually:
also have \[ \ldots \leq 2 \ast C \ast \pi / R + 2 \ast C \ast \pi \ast (1 / N + 1 / R) + 3 \ast M / R / \ln N + 6 \ast R \ast M \ast N \] proof (rule order.trans[OF norm-triangle-ineq] add-mono)+
have \[ \oint [B1] g = - \oint [reversepath B1] g \] by (simp add: contour-integral-reversepath)
also have \[ \oint [reversepath B1] g = \text{integral} \{- \delta..0\} (\lambda x. g (x + R \ast i)) \]
unfolding B1-def reversepath-linepath using \( \delta \)
by (subst contour-integral-linepath-same-Im) auto
also have \[ \text{norm} (- \ldots) = \text{norm} \ldots \] by simp
also have \[ \ldots \leq 3 \ast M / R / \ln N \] using \( R \) by (intro B12-bound) auto
finally show \[ \text{norm} (\oint [B1] g) \leq \ldots . \]
next have \[ \oint [B3] g = \text{integral} \{- \delta..0\} (\lambda x. g (x + (-R) \ast i)) \]
unfolding B3-def
using \( \delta \)
by (subst contour-integral-linepath-same-Im) auto
also have \[ \text{norm} \ldots \leq 3 \ast M / R / \ln N \] using \( R \) by (intro B12-bound)

auto
finally show \[ \text{norm} (\oint [B3] g) \leq \ldots . \]
next have \[ \text{norm} (\oint [B2] g) \leq M \ast N \] powr \(- \delta \ast (3 / \delta) \ast \)
\[ \text{norm} (\text{Complex} (- \delta) (-R) - \text{Complex} (- \delta) \ast R) \] unfolding B2-def
proof ((rule contour-integral-bound-linepath; (fold B2-def)?), goal-cases)
\begin{enumerate}
\item \( 3 R \) have \[ \text{simp}: z \neq 0 \] and \( \text{Re:z} \) \( \text{Re z} = - \delta \) and \( \text{Im:z} \) \( \text{Im z} \in \{-\delta..R\} \)
by (auto simp: closed-segment-same-Im closed-segment-eq-real-ivl)
\item \( 3 \) have \[ z \in X' \] using \( R \delta \) path-images by (auto simp: B2-def)
\item \( 3 \delta R \) have \[ \text{norm} z \leq \text{sqr}t (\delta^2 + R^2) \] unfolding cmod-def using \( \text{Re:z} \) \( \text{Im:z} \)
by (intro real-sqr:le-mono add-mono) (auto simp: power2-le-iff-abs-le)
\end{enumerate}
from \[ \text{power-mono}[\text{OF this, of 2}] \] have \[ \text{norm-sqr: norm} z \ast 2 \leq \delta^2 + R^2 \]
by simp
have \[ \text{norm} (1 / z + z / R^2) \leq (1 + \text{norm} z)^2 / R^2 \] / \( \delta \)
unfolding add-divide-distrib using \( \delta \) \( R \) abs-Re-le-cmod[\text{of z}]

58
by (intro order_trans[OF norm-triangle-ineq] add-mono)
  (auto simp: norm-divide norm-mult field-simps power2_eq_square Re-z)
also have \( \leq (1 + (1 + \delta^2 / R^2)) / \delta \) using \( \delta R \) \( (z \in X') \) norm-sqr
unfolding X'-def
by (intro divide-right-mono add-left-mono)
  (auto simp: field-simps in-cbox-complex-iff intro!: power mono)
also have \( \delta^2 / R^2 \leq 1 \)
using \( \delta R \) by (auto simp: field-simps intro!: power mono)
finally have \( \text{norm} (1 / z + z / R^2) \leq 3 / \delta \) using \( \delta \)
  (simp add: divide-right-mono)
with \( (z \in X') \) show \( \text{norm} (g z) \leq M * N \text{ powr} (-\delta) * (3 / \delta) \)
unfolding g-def norm-mult
by (intro mult_mono mult_nonneg_nonneg M) (auto simp: norm_power_real_pow)
rewrites

qed (insert path-images M \( \delta \), auto intro!: contour_integrable_holomorphic_simple[OF g_holo])

thus \( \text{norm} (\{ [B2] g \}) \leq 6 * R * M * N \text{ powr} (-\delta) / \delta \)
using \( R \) by (simp add: field_simps cmod_def real_sqrt_mult)

next
have \( \text{norm} (\{ [A] (\lambda x. g-S (- x))) \leq (2 * C / (\text{real} N * R) + 2 * C / R^2) * \)
\( R \) * \((\pi/2) - (-\pi/2) \)
unfolding A-def
proof ((rule contour_integral_bound_part_circlepath_strong)[where \( k = \{ R * i, -R * i \} \);]
  (fold A-def) ?, goal_cases)
case (0 z)
hence [simp]: \( z \neq 0 \) and \( \text{norm} z = R \) using \( R \)
by (auto simp: A-def dest!: in_path_image_part_circlepath)
from 0 have \( \text{Re} z \neq 0 \)
using (norm z = R) by (auto simp: cmod_def abs_if complex_eq_iff split: if_splits)

with 0 have \( \text{Re} z > 0 \) using image-A by auto
have \( S (w - z) = (\sum k = 1..N. fds_nth F k / of_nat k \text{ powr} (w - z)) \)
  by (simp add: S_def eval_fds_truncate)
also have \( \text{norm} \ldots \leq C * N \text{ powr} \text{Re} z * (1 / N + 1 / \text{Re} z) \)
using \( \text{Re} z > 0 \) \( w N \) by (intro newman-ingham-again2 C) auto
finally have \( \text{norm} (S (w - z)) \leq \ldots . \)
  hence \( \text{norm} (g-S (-z)) \leq \)
  \((2 * N \text{ powr} (\text{Re} z) * (1 / N + 1 / \text{Re} z)) * N \text{ powr} (-\text{Re} z) * \)
\((2 * \text{Re} z / R^2) \)
unfolding g-S-def norm_mult
using newman-ingham-again2[OF \( \langle \text{norm} z = R \rangle \) \( \langle \text{Re} z > 0 \rangle \) \( \langle C \geq 1 \rangle \) \( R \)]
by (intro mult_mono mult_nonneg_nonneg circle_bound)
(auto simp: norm_power_real_pow norm_minus_minus)
also have \( \ldots = 2 * C * (\text{Re} z / N + 1) / R^2 \) using \( R N \) \( \langle \text{Re} z > 0 \rangle \)
by (simp add: power_minus algebra_simps)
also have \( \ldots < 2 * C / (N * R) + 2 * C / R^2 \)
unfolding add_divide_distrib
ring_distrib
using \( \text{Re} z \) \( \text{abs-Real-cmod} \) of z \( \langle \text{norm} z = R \rangle \) \( \langle \text{Re} z > 0 \rangle \) \( \langle C \geq 1 \rangle \)
by (intro add-mono) (auto simp: power2-eq-square field-simps mult-mono)  

finally show \( ? \) case .  

qed {insert \( R \) \( image-A \) \( C \), auto intro!: contour-integrable-holomorphic-simple[of - \{0\}]  
holomorphic-intros simp: g-S-def S-def)  
also have \( \ldots \) = \( 2 \cdot C \cdot pi \cdot (1 / N + 1 / R) \) using \( R \) \( N \)  
by (simp add: power2-eq-square field-simps)  

finally show \( \text{norm} \ (\lambda x. g-S (- x)) \) \( \leq \ldots . \)  

next  

have \( \text{norm} \ (\lambda x. g-rem) \) \( \leq (2 \cdot C \cdot R^2) \cdot R \cdot ((pi/2) - (-pi/2)) \)  
unfolding A-def  

proof {  
(rule contour-integral-bound-part-circlepath-strong[where \( k = \{ R \) \( * \) \( i, -R+i\} \})\);  
(fold A-def)\};  
goal-cases  
case \( 6 \) \( z \)  

hence [simp]: \( z \neq 0 \) and \( \text{norm} \ z = R \) using \( R \)  
by (auto simp: A-def dest: in-path-image-part-circlepath)  
from \( 6 \) have \( \text{Re} \ z \neq 0 \)  
using \( \text{norm} \ z = R \) by (auto simp: cmod-def abs-if complex-eq-iff split: if-splits)  

with \( 6 \) have \( \text{Re} \ z > 0 \) using \( image-A \) by auto  

have summable: summable \( (\lambda n. C * (1 / (Suc n + N) \text{ powr} \ (Re w + Re z))) \)  
using summable-hurwitz-zeta-real[of \( Re \) \( w \) \( + \) \( Re \) \( z \) \( Suc \) \( N \) \( \langle \text{Re} \ z > 0 \rangle \) \( w \)  
unfolding powr-minus by (intro summable-mult) (auto simp: field-simps)  
have rem \( (w + z) = (\sum n. fds-nth F \ (Suc n + N) / \ (Suc n + N) \text{ powr} \ (w + z)) \)  
using \( \text{Re} \ z > 0 \) \( w \) by (simp add: rem-altdef eval-fds-remainder)  
also have \( \text{norm} \ \ldots \ \leq (\sum n. C / \ (Suc n + N) \text{ powr} \ Re \ (w + z)) \) using summable  
by (intro norm-suminf-le)  
(auto simp: norm-divide norm-powr-powr intro!: divide-right-mono C)  

also have \( \ldots = (\sum n. C * \ (Suc n + N) \text{ powr} \ - \text{Re} \ (w + z)) \)  
unfolding powr-minus by (simp add: field-simps)  
also have \( \ldots = C * (\sum n. Suc n + N) \text{ powr} \ - \text{Re} \ (w + z)) \)  
using summable-hurwitz-zeta-real[of \( Re \ w \) \( + \) \( Re \ z \) \( Suc \) \( N \) \( \langle \text{Re} \ z > 0 \rangle \) \( w \)  
by (subst suminf-mult) (auto simp: add-ac)  
also have \( (\sum n. Suc n + N) \text{ powr} \ - \text{Re} \ (w + z) \) \( \leq \)  
\( N \text{ powr} \ (1 - \text{Re} \ (w + z)) / \ (\text{Re} \ (w + z) - 1) \)  
using \( \langle \text{Re} \ z > 0 \rangle \) \( w \) \( N \) hurwitz-zeta-real-bound-aux[of \( N \) \( \text{Re} \ (w + z) \)]  
by (auto simp: add-ac)  
also have \( \ldots \ \leq N \text{ powr} \ - \text{Re} \ z / \text{Re} \ z \)  
using \( w \) \( N \) \( \langle \text{Re} \ z > 0 \rangle \) by (intro frac-le powr-mono) auto  
finally have \( \text{norm} \ (\text{rem} \ (w + z)) \) \( \leq C / \ (\text{Re} \ z * N \text{ powr} \ \text{Re} \ z) \)  
using C by (simp add: mult-left-zeroe mult-right-zeroe powr-minus field-simps)  

hence \( \text{norm} \ (g-rem z) \ \leq (C / \ (\text{Re} \ z * N \text{ powr} \text{Re} \ z)) * N \text{ powr} \text{Re} \ z *
(2 * Re z / R^2)

unfolding g-rem-def norm-mult
using newman-ingham-aux1[OF - (norm z = R) R (Re z > 0) C
by (intro mult-mono mult-nonneg-nonneg circle-bound)
(auto simp: norm-powr-real-powr norm-uminus-minus)
also have ... = 2 * C / R^2 using R N (Re z > 0)
by (simp add: power-minus field-simps)
finally show ?case .

next
show g-rem contour-integrable-on A using path-images
by (auto simp: g-rem-def rem-def S-def intro: contour-integrable-holomorphic-simple[of - X -{0}]
holomorphic-intros)
qed (insert R N C, auto)
also have ... = 4*C*pi/R + 2*C*pi/N + 6*M/R / ln N + 6*R*M*N powr - δ / δ
by (simp add: algebra-simps)
also have ... = 2*pi * (2*C/R + C/N + 3*M / (pi*R*ln N) + 3*R*M / (δ*pi * N powr δ))
by (simp add: field-simps power-minus )
also have norm (2 * pi * i * (f w - S w)) = 2 * pi * norm (f w - S w)
by (simp add: norm-mult)
finally have norm (f w - S w) ≤ bound N by (simp add: bound-def)
also have bound N < ε by fact
finally show norm (f w - S w) < ε .
qed

thus fds-converges F w
by (auto simp: fds-converges-altdef2 intro: convergentI)
thus eval-fds F w = f w
using (λN. eval-fds (fds-truncate N F) w) −→ f w
by (intro tendsto-unique[OF - tendsto-eval-fds-truncate]) auto

end

3 Prime-Counting Functions

theory Prime-Counting-Functions
  imports Prime-Number-Theorem-Library
begin

We will now define the basic prime-counting functions π, ϑ, and ψ. Additionally, we shall define a function M that is related to Mertens theorems and Newmans proof of the Prime Number Theorem. Most of the results in
this file are not actually required to prove the Prime Number Theorem, but
are still nice to have.

3.1 Definitions

definition prime-sum-upto :: (nat ⇒ 'a) ⇒ real ⇒ 'a :: semiring_1 where
  prime-sum-upto f x = (∑ p | prime p ∧ real p ≤ x. f p)

lemma prime-sum-upto-altdef1:
  prime-sum-upto f x = sum-upto (λ. ind prime p * f p) x
unfolding sum-upto-def prime-sum-upto-def
by (intro sum.mono-neutral-cong-left finite-subset[of - finite-Nats-le-real[of x]])
  (auto dest: prime-gt-1-nat simp: ind-def)

lemma prime-sum-upto-altdef2:
  prime-sum-upto f x = (∑ p | prime p ∧ p ≤ nat ⌊x⌋. f p)
unfolding sum-upto-altdef prime-sum-upto-altdef1
by (intro sum.mono-neutral-cong-right) (auto simp: ind-def dest: prime-gt-1-nat)
lemma prime-sum-upto-altdef3:
  prime-sum-upto f x = (∑ p ← primes-upto (nat ⌊x⌋). f p)
proof
  have (∑ p←primes-upto (nat ⌊x⌋). f p) = (∑ p | prime p ∧ p ≤ nat ⌊x⌋. f p)
    using assms by (intro sum-list-distinct-conv-sum-set) (auto simp: set-primes-upto conj-commute)
thus ?thesis by (simp add: prime-sum-upto-altdef2)
qed

lemma prime-sum-upto-eqI:
  assumes a ≤ b ⋀ k ∈ {nat ⌊a⌋<..<nat ⌊b⌋} =⇒ ¬prime k
  shows prime-sum-upto f a = prime-sum-upto f b
proof
  have *: k ≤ nat ⌊a⌋ if k ≤ nat ⌊b⌋ prime k for k
    using that assms(2)[of k] by (cases k ≤ nat ⌊a⌋) auto
  from assms(1) have nat ⌊a⌋ ≤ nat ⌊b⌋ by linarith
  hence (∑ p | prime p ∧ p ≤ nat ⌊a⌋. f p) = (∑ p | prime p ∧ p ≤ nat ⌊b⌋. f p)
    using assms by (intro sum.mono-neutral-left) (auto dest: *)
thus ?thesis by (simp add: prime-sum-upto-altdef2)
qed

lemma prime-sum-upto-eqI':
  assumes a' ≤ nat ⌊a⌋ a ≤ b nat ⌊b⌋ ≤ b' ⋀ k. k ∈ {a'<..<b'} =⇒ ¬prime k
  shows prime-sum-upto f a = prime-sum-upto f b
by (rule prime-sum-upto-eqI) (use assms in auto)

lemmas eval-prime-sum-upto = prime-sum-upto-altdef3[unfolded primes-upto-sieve]

lemma of-nat-prime-sum-upto: of-nat (prime-sum-upto f x) = prime-sum-upto (λp. of-nat (f p)) x
  by (simp add: prime-sum-upto-def)
lemma prime-sum-upto-mono:
assumes \( \forall n. n > 0 \implies f n \geq (\theta :: \text{real}) \) \( x \leq y \)
shows \( \text{prime-sum-upto} f x \leq \text{prime-sum-upto} f y \)
using assms unfolding prime-sum-upto-altdef1 sum-upto-altdef
by (intro sum-mono2) (auto simp: le_nat_iff le_floor_iff ind-def)

lemma prime-sum-upto-nonneg:
assumes \( \forall n. n > 0 \implies f n \geq (\theta :: \text{real}) \)
shows \( \text{prime-sum-upto} f x \geq 0 \)
unfolding prime-sum-upto-altdef1 sum-upto-altdef
by (intro sum-nonneg) (auto simp: ind-def assms)

lemma prime-sum-upto-eq-0:
assumes \( x < 2 \)
shows \( \text{prime-sum-upto} f x = 0 \)
proof
  from assms have \( \text{nat } \lfloor x \rfloor = 0 \lor \text{nat } \lfloor x \rfloor = 1 \) by linarith
  thus ?thesis by (auto simp: eval-prime-sum-upto)
qed

lemma measurable-prime-sum-upto [measurable]:
fixes \( f :: 'a \Rightarrow \text{nat} \Rightarrow \text{real} \)
assumes \( \forall y. (\lambda t. f t y) \in M \rightarrow M \) borel
assumes \( x \in M \rightarrow M \) borel
shows \( (\lambda t. \text{prime-sum-upto} (f t) (x t)) \in M \rightarrow M \) borel
unfolding prime-sum-upto-altdef1 by measurable

The following theorem breaks down a sum over all prime powers no greater
than fixed bound into a nicer form.

lemma sum-upto-primepows:
fixes \( f :: \text{nat} \Rightarrow 'a :: \text{comm-monoid-add} \)
assumes \( \forall n. \neg \text{primepow } n \implies f n = 0 \) \( \forall i. \text{prime } p \implies i > 0 \implies f (p ^ i) = g p i \)
shows \( \text{sum-upto} f x = (\sum (p, i) | \text{prime } p \land i > 0 \land \text{real } (p ^ i) \leq x. g p i) \)
proof
  let \( \text{aprimedivisor} \)
  have \( g : g (?d n) \) (multiplicity (?d n)) = \( f n \) if primepow n for n using that
    by (subst assms(2) [symmetric])
    (auto simp: primepow-decompose aprimedivisor-prime-power primepow-gt-Suc-0
    intro!: aprimedivisor-nat multiplicity-aprimedivisor-gt-0-nat)
  have \( \text{sum-upto} f x = (\sum n | \text{primepow } n \land \text{real } n \leq x. f n) \)
  unfolding sum-upto-def using assms
  also have \( \ldots = (\sum (p, i) | \text{prime } p \land i > 0 \land \text{real } (p ^ i) \leq x. g p i) \) (is - = \( \sum - ?S \))
    by (rule sum.reindex-bij-witness[of - \lambda (p,i). p ^ i \land n. (?d n, multiplicity (?d n))])
    (auto simp: aprimedivisor-prime-power primepow-decompose primepow-gt-Suc-0

63
simp del: of-nat-power intro!: aprime-divisor-nat multiplicity-aprime-divisor-gt-0-nat

finally show ?thesis.

qed

definition primes-pi where primes-pi = prime-sum-upto (λ p. 1 :: real)
definition primes-theta where primes-theta = prime-sum-upto (λ p. ln (real p))
definition primes-psi where primes-psi = sum-upto (mangoldt :: nat ⇒ real)
definition primes-M where primes-M = prime-sum-upto (λ p. ln (real p) / real p)

Next, we define some nice optional notation for these functions.

bundle prime-counting-notation
begin

notation primes-pi (π)
notation primes-theta (ϑ)
notation primes-psi (ψ)
notation primes-M (ℜ)

end

bundle no-prime-counting-notation
begin

no-notation primes-pi (π)
nor-notation primes-theta (ϑ)
nor-notation primes-psi (ψ)
nor-notation primes-M (ℜ)

end

lemmas π-def = primes-pi-def
lemmas ϑ-def = primes-theta-def
lemmas ψ-def = primes-psi-def

lemmas eval-π = primes-pi-def [unfolded eval-prime-sum-upto]
lemmas eval-ϑ = primes-theta-def [unfolded eval-prime-sum-upto]
lemmas eval-ℜ = primes-M-def [unfolded eval-prime-sum-upto]

3.2 Basic properties

The proofs in this section are mostly taken from Apostol [1].

lemma measurable-π [measurable]: π ∈ borel →_M borel
and measurable-ϑ [measurable]: ϑ ∈ borel →_M borel
and measurable-ψ [measurable]: ψ ∈ borel →_M borel
and measurable-primes-M [measurable]: ℜ ∈ borel →_M borel
unfolding primes-M-def π-def ϑ-def ψ-def by measurable
lemma \( \pi\text{-eq-0} \) [simp]: \( x < 2 \implies \pi x = 0 \)
and \( \vartheta\text{-eq-0} \) [simp]: \( x < 2 \implies \vartheta x = 0 \)
and \( \text{primes-M}\text{-eq-0} \) [simp]: \( x < 2 \implies \mathfrak{M} x = 0 \)
unfolding primes-pi-def primes-theta-def primes-M-def
by (rule prime-sum-upto-eq-0; simp)+

lemma \( \pi\text{-nat-cancel} \) [simp]: \( \pi (\text{nat } x) = \pi x \)
and \( \vartheta\text{-nat-cancel} \) [simp]: \( \vartheta (\text{nat } x) = \vartheta x \)
and \( \text{primes-M}\text{-nat-cancel} \) [simp]: \( \mathfrak{M} (\text{nat } x) = \mathfrak{M} x \)
and \( \psi\text{-nat-cancel} \) [simp]: \( \psi (\text{nat } x) = \psi x \)
and \( \text{primes-M-floor-cancel} \) [simp]: \( \mathfrak{M} (\text{of-int } \lfloor y \rfloor) = \mathfrak{M} y \)
and \( \psi\text{-floor-cancel} \) [simp]: \( \psi (\text{of-int } \lfloor y \rfloor) = \psi y \)
by (simp-all add: \( \pi\text{-def} \) \( \vartheta\text{-def} \) \( \psi\text{-def} \) primes-M-def prime-sum-upto-altdef2 sum-upto-altdef)

lemma \( \pi\text{-nonneg} \) [intro]: \( \pi x \geq 0 \)
and \( \vartheta\text{-nonneg} \) [intro]: \( \vartheta x \geq 0 \)
and \( \text{primes-M}\text{-nonneg} \) [intro]: \( \mathfrak{M} x \geq 0 \)
unfolding primes-pi-def primes-theta-def primes-M-def
by (rule prime-sum-upto-nonneg; simp)+

lemma \( \pi\text{-mono} \) [intro]: \( x \leq y \implies \pi x \leq \pi y \)
and \( \vartheta\text{-mono} \) [intro]: \( x \leq y \implies \vartheta x \leq \vartheta y \)
and \( \text{primes-M}\text{-mono} \) [intro]: \( x \leq y \implies \mathfrak{M} x \leq \mathfrak{M} y \)
unfolding primes-pi-def primes-theta-def primes-M-def
by (rule prime-sum-upto-mono; simp)+

lemma \( \pi\text{-pos-iff} \): \( \pi x > 0 \iff x \geq 2 \)
proof
assume \( x \): \( x \geq 2 \)
show \( \pi x > 0 \)
by (rule less-le-trans[OF - \( \pi\text{-mono} \) \( \text{OF } x \)]) (auto simp: eval-\( \pi \))
next
assume \( \pi x > 0 \)
hence \( \neg(x < 2) \) by auto
thus \( x \geq 2 \) by simp
qed

lemma \( \pi\text{-pos} \): \( x \geq 2 \implies \pi x > 0 \)
by (simp add: \( \pi\text{-pos-iff} \))

lemma \( \psi\text{-eq-0} \) [simp]:
assumes \( x < 2 \)
shows \( \psi x = 0 \)
proof
from assms have \( \text{nat } [x] \leq 1 \) by linarith
hence \( \text{mangoldt } n = (0 :: \text{real}) \) if \( n \in \{0 < \text{nat } [x] \} \) for \( n \)
using that by (auto simp: mangoldt-def dest!: primepow-gt-Suc-0)
thus thesis unfolding ψ-def sum-upto-altdef by (intro sum.neutral) auto
qed

lemma ψ-nonneg [intro]: ψ x ≥ 0
  unfolding ψ-def sum-upto-def by (intro sum-nonneg mangoldt-nonneg)

lemma ψ-mono: x ≤ y ⇒ ψ x ≤ ψ y
  unfolding ψ-def sum-upto-def by (intro sum-mono2 mangoldt-nonneg) auto

3.3 The n-th prime number

Next we define the n-th prime number, where counting starts from 0. In traditional mathematics, it seems that counting usually starts from 1, but it is more natural to start from 0 in HOL and the asymptotics of the function are the same.

definition nth-prime :: nat ⇒ nat where
  nth-prime n = (THE p. prime p ∧ card {q. prime q ∧ q < p} = n)

lemma finite-primes-less [intro]: finite {q::nat. prime q ∧ q < p}
  by (rule finite-subset[of - {..<p}]) auto

lemma nth-prime-unique-aux:
  fixes p p' :: nat
  assumes prime p card {q. prime q ∧ q < p} = n
  assumes prime p' card {q. prime q ∧ q < p'} = n
  shows p = p'
  using assms
  proof (induction p p' rule: linorder-wlog)
    case (le p p')
    have finite {q. prime q ∧ q < p'} by (rule finite-primes-less)
    moreover from le have {q. prime q ∧ q < p} ⊆ {q. prime q ∧ q < p'}
      by auto
    moreover from le have card {q. prime q ∧ q < p} = card {q. prime q ∧ q < p'}
      by simp
    ultimately have {q. prime q ∧ q < p} = {q. prime q ∧ q < p'}
      by (rule card-subset-eq)
    with prime p have ¬(p < p') by blast
    with p ≤ p' show p = p' by auto
  qed auto

lemma π-smallest-prime-beyond:
  π (real (smallest-prime-beyond m)) = π (real (m - 1)) + 1
  proof (cases m)
    case 0
    have smallest-prime-beyond 0 = 2
      by (rule smallest-prime-beyond-eq) (auto dest: prime-gt-1-nat)
with \(\theta\) show \(?\)thesis by (simp add: eval-π)

next

case (Suc \(n\))
define \(n'\) where \(n' = \text{smallest-prime-beyond} (\text{Suc} \ n)\)

have \(n < n'\)
  using \(\text{smallest-prime-beyond-le}[\text{of} \ \text{Suc} \ n]\) unfolding \(n'\)-def by linarith

have \(\text{prime} \ n'\) by (simp add: \(n'\)-def)

have \(n' \leq p\) if \(\text{prime} \ p\) \(p > n\) for \(p\)
  using that \(\text{smallest-prime-beyond-smallest}[\text{of} \ p \ \text{Suc} \ n]\)
  by (auto simp: \(n'\)-def)

note \(n' = (n < n') (\text{prime} \ n')\) this

have \(\pi (\text{real} \ n') = \text{real} (\text{card} \ \{p. \ \text{prime} \ p \land p \leq n'\})\)
  by (simp add: \(\pi\)-def prime-sum-upto-def)

also have \(\text{Suc} \ n \leq \text{n'}\) unfolding \(n'\)-def by (rule \(\text{smallest-prime-beyond-le}\))

hence \(\{p. \ \text{prime} \ p \land p \leq n'\} = \{p. \ \text{prime} \ p \land p \leq n\} \cup \{p. \ \text{prime} \ p \land p \in \{n < ... n'\}\}\)
  by auto

also have \(\text{real} (\text{card} \ldots) = \pi (\text{real} \ n) + \text{real} (\text{card} \ \{p. \ \text{prime} \ p \land p \in \{n < ... n'\}\})\)
  by (simp: \(\pi\)-def prime-sum-upto-def)

also have \(\{p. \ \text{prime} \ p \land p \in \{n < ... n'\}\} = \{n'\}\)
  using \(n'\) by (auto intro: antisym)

finally show \(?\)thesis using \(\text{Suc}\) by (simp add: \(n'\)-def)

qed

lemma \(\pi\)-inverse-exists: \(\exists n. \ \pi (\text{real} \ n) = \text{real} m\)
proof (induction \(m\))

case \(0\)
show \(?\)case by (intro exI[of - 0]) auto

next

case (Suc \(m\))

from \(\pi\)-IH obtain \(n\) where \(\pi (\text{real} \ n) = \text{real} m\)
  by auto

hence \(\pi (\text{real} (\text{smallest-prime-beyond} (\text{Suc} \ n))) = \text{real} (\text{Suc} \ m)\)
  by (subst \(\pi\)-smallest-prime-beyond) auto

thus \(?\)case by blast

qed

lemma \(n\)th-prime-exists: \(\exists p::\text{nat}. \ \text{prime} \ p \land \text{card} \ \{q. \ \text{prime} q \land q < p\} = n\)
proof

from \(\pi\)-inverse-exists[of \(n\)] obtain \(m\) where \(\pi (\text{real} \ m) = \text{real} n\) by blast

hence \(\text{card} : \text{card} \ \{q. \ \text{prime} q \land q \leq m\} = n\)
  by (auto simp: \(\pi\)-def prime-sum-upto-def)

define \(p\) where \(p = \text{smallest-prime-beyond} (\text{Suc} \ m)\)

have \(m < p\) using \(\text{smallest-prime-beyond-le}[\text{of} \ \text{Suc} \ m]\) unfolding \(p\)-def by linarith

have \(\text{prime} \ p\) by (simp add: \(p\)-def)

have \(p \leq q\) if \(\text{prime} \ q \land q > m\) for \(q\)
  using \(\text{smallest-prime-beyond-smallest}[\text{of} \ q \ \text{Suc} \ m]\) that by (simp add: \(p\)-def)

67
\[ n = \langle m < p \rangle \langle \text{prime } p \rangle \text{ this} \]

have \( \{ q \text{. prime } q \land q < p \} = \{ q \text{. prime } q \land q \leq m \} \)
proof safe
  fix \( q \) assume \( \text{prime } q \land q < p \)
  hence \( \neg(q > m) \) using \( p(1,2) \leftrightarrow p(3)\{\text{of } q\} \) by auto
  thus \( q \leq m \) by simp
qed (insert \( p \), auto)
also have card \( \ldots = n \) by fact
finally show \( \text{thesis} \) using \( \langle \text{prime } p \rangle \) by blast
qed

lemma nth-prime-exists1: \( \exists! p :: \text{nat} \text{. prime } p \land \text{card } \{ q \text{. prime } q \land q < p \} = n \)
by (intro ex-ex1I nth-prime-exists) (blast intro: nth-prime-unique-aux)

lemma prime-nth-prime [intro]: \( \text{prime } (\text{nth-prime } n) \)
and card-less-nth-prime [simp]: \( \text{card } \{ q \text{. prime } q \land q < \text{nth-prime } n \} = n \)
using the1' [OF nth-prime-exists1[of \( n \)]] by (simp-all add: nth-prime-def)

lemma card-le-nth-prime [simp]: \( \text{card } \{ q \text{. prime } q \land q \leq \text{nth-prime } n \} = \text{Suc } n \)
proof -
  have \( \{ q \text{. prime } q \land q \leq \text{nth-prime } n \} = \text{insert } p \{ q \text{. prime } q \land q < \text{nth-prime } n \} \)
    by auto
  also have card \( \ldots = \text{Suc } n \) by simp
  finally show \( \text{thesis} \).
qed

lemma \( \pi \text{-nth-prime} \) [simp]: \( \pi (\text{real } (\text{nth-prime } n)) = \text{real } n + 1 \)
by (simp add: \( \pi \)-def prime-sum-upto-def)

lemma nth-prime-eqI:
  assumes \( \text{prime } p \land \text{card } \{ q \text{. prime } q \land q < p \} = n \)
  shows \( \text{nth-prime } n = p \)
unfolding nth-prime-def
by (rule the1-equality[OF nth-prime-exists1]) (use assms in auto)

lemma nth-prime-eqI':
  assumes \( \text{prime } p \land \text{card } \{ q \text{. prime } q \land q \leq p \} = \text{Suc } n \)
  shows \( \text{nth-prime } n = p \)
proof (rule nth-prime-eqI)
  have \( \{ q \text{. prime } q \land q \leq p \} = \text{insert } p \{ q \text{. prime } q \land q < p \} \)
    using assms by auto
  also have card \( \ldots = \text{Suc } (\text{card } \{ q \text{. prime } q \land q < p \}) \)
    by simp
  finally show card \( \{ q \text{. prime } q \land q < p \} = n \) using assms by simp
qed (use assms in auto)

lemma nth-prime-eqI'':

68
assumes prime p π (real p) = real n + 1
shows nth-prime n = p
proof (rule nth-prime-eqI)
  have real (card {q. prime q ∧ q ≤ p}) = π (real p)
    by (simp add: π-def prime-sum-upto-def)
  also have ... = real (Suc n) by (simp add: assms)
  finally show card {q. prime q ∧ q ≤ p} = Suc n
    by (simp only: of-nat-eq-iff)
qed fact+

lemma nth-prime-0 [simp]: nth-prime 0 = 2
  by (intro nth-prime-eqI) (auto dest: prime-gt-1-nat)

lemma nth-prime-Suc: nth-prime (Suc n) = smallest-prime-beyond (Suc (nth-prime n))
  by (rule nth-prime-eqI) (simp-all add: π-smallest-prime-beyond)

lemmas nth-prime-code [code] = nth-prime-0 nth-prime-Suc

lemma strict-mono-nth-prime: strict-mono nth-prime
proof (rule strict-monoI-Suc)
  fix n :: nat
  have Suc (nth-prime n) ≤ smallest-prime-beyond (Suc (nth-prime n)) by simp
  also have ... = nth-prime (Suc n) by (simp add: nth-prime-Suc)
  finally show nth-prime n < nth-prime (Suc n) by simp
qed

lemma nth-prime-le-iff [simp]: nth-prime m ≤ nth-prime n ↔ m ≤ n
  using strict-mono-le-iff[OF strict-mono-nth-prime] by blast

lemma nth-prime-less-iff [simp]: nth-prime m < nth-prime n ↔ m < n
  using strict-mono-less[OF strict-mono-nth-prime] by blast

lemma nth-prime-eq-iff [simp]: nth-prime m = nth-prime n ↔ m = n
  using strict-mono-eq[OF strict-mono-nth-prime] by blast

lemma nth-prime-ge-2: nth-prime n ≥ 2
  using nth-prime-le-iff[of 0 n] by (simp del: nth-prime-le-iff)

lemma nth-prime-lower-bound: nth-prime n ≥ Suc (Suc n)
proof
  have n = card {q. prime q ∧ q < nth-prime n}
  by simp
  also have ... ≤ card {2..<nth-prime n}
  by (intro card_mono) (auto dest: prime-gt-1-nat)
  also have ... = nth-prime n - 2 by simp
  finally show ?thesis using nth-prime-ge-2[of n] by linarith
qed
proof (rule filterlim-at-top-mono)
  show \( \text{filterlim} (\lambda n::\text{nat}. n + 2) \text{ at-top at-top by } \text{real-asym} \)
qed (auto simp: nth-prime-lower-bound)

3.4 Relations between different prime-counting functions

The \( \psi \) function can be expressed as a sum of \( \vartheta \).

lemma \( \psi\text{-altdef} \):
  assumes \( x > 0 \)
  shows \( \psi x = \text{sum-upto} (\lambda m. \text{prime-sum-upto} \ln (\text{root} m x)) (\log 2 x) \) (is \( \sim \) ?rhs)
proof
  have finite: finite \( \{ p. \text{prime} p \land \text{real} p \leq y \} \) for \( y \)
    by (rule finite-subset[of \{-..nat \{y\}\}] (auto simp: le-nat-iff le-floor-iff)
  define \( S \) where \( S = (\Sigma i. \{ i. 0 < i \land \text{real} i \leq \log 2 x \}. \{ p. \text{prime} p \land \text{real} p \leq \text{root} i x \}) \)
  have \( \psi x = (\sum (p, i) \mid \text{prime} p \land 0 < i \land \text{real} (p \circ i) \leq x. \ln (\text{real} p)) \) unfolding \( \psi\text{-def} \)
    by (subst sum-upto-primepows[where \( g = \lambda p i. \ln (\text{real} p) \)])
      (auto simp: case-prod-unfold mangoldt-non-primepow)
  also have \( \ldots = (\sum (i, p) \mid \text{prime} p \land 0 < i \land \text{real} (p \circ i) \leq x. \ln (\text{real} p)) \)
    by (intro sum.reindex-bij-witness[of - \( \lambda (x,y). (y,x) \lambda(x,y). (y,x) \)] auto
  also have \( \{(i, p). \text{prime} p \land 0 < i \land \text{real} (p \circ i) \leq x \} = S \)
    unfolding \( S\text{-def} \)
proof safe
  fix \( i p :: \text{nat} \) assume \( ip; i > 0 \) \( \text{real} i \leq \log 2 x \) \( \text{prime} p \) \( \text{real} p \leq \text{root} i x \)
  hence \( \text{real} (p \circ i) \leq \text{root} i x \) by \( \text{real-power} \)
  with \( ip \) \( \text{assms} \) show \( \text{real} (p \circ i) \leq x \) by \text{simp}
next
  fix \( i p \) assume \( ip; \text{prime} p \) \( i > 0 \) \( \text{real} (p \circ i) \leq x \)
  from \( ip \) have \( 2 \circ i \leq p \circ i \) by \( \text{real-power} \) (auto dest: prime-gt-1-nat)
  also have \( \ldots \leq x \) using \( ip \) by \text{simp}
  finally show \( \text{real} i \leq \log 2 x \)
    using \( \text{assms} \) by \( \text{simp add: le-log powr-power} \)
  have \( \text{root} i (\text{real} p \circ i) \leq \text{root} i x \) using \( ip \) \( \text{assms} \)
    by (subst real-root-le-iff) auto
  also have \( \text{root} i (\text{real} p \circ i) = \text{real} p \)
    using \( \text{assms} \) \( \text{ip} \) by (subst real-root-pos2) auto
  finally show \( \text{real} p \leq \text{root} i x \).
qed

also have \( (\sum (i,p) \in S. \ln p) = \text{sum-upto} (\lambda m. \text{prime-sum-upto} \ln (\text{root} m x)) (\log 2 x) \)
  unfolding \( \text{sum-upto-def prime-sum-upto-def} \) \( S\text{-def} \) using \( \text{finite} \) by (subst sum Sigma) auto
  finally show \( ?\text{thesis} \).
qed
lemma \(\psi\)-conv-\(\vartheta\)-sum: \(x > 0 \rightarrow \psi x = \text{sum-upto} (\lambda m. \vartheta (\text{root } m x)) (\log 2 x)\)
by (simp add: \(\psi\)-altdef \(\vartheta\)-def)

lemma \(\psi\)-minus-\(\vartheta\):
assumes \(x: x \geq 2\)
shows \(\psi x - \vartheta x = (\sum i \mid 2 \leq i \land \text{real } i \leq \log 2 x) \vartheta (\text{root } i x)\)
proof
- have \(\text{finite}: \text{finite} \{i. 2 \leq i \land \text{real } i \leq \log 2 x\}\)
  by (rule finite-subset[of \{2..nat \{\log 2 x\}\}]) (auto simp: le-nat-iff le-floor-iff)
  have \(\psi x = (\sum i \mid 0 < i \land \text{real } i \leq \log 2 x) \vartheta (\text{root } i x)\) using \(x\)
  by (simp add: \(\psi\)-conv-\(\vartheta\)-sum sum-upto-def)
also have \(\{i. 0 < i \land \text{real } i \leq \log 2 x\} = \text{insert } 1 \{i. 2 \leq i \land \text{real } i \leq \log 2 x\}\)
using \(x\)
  by (auto simp: le-log-iff)
also have \((\sum i \in\ldots \vartheta (\text{root } i x)) - \vartheta x =
(\sum i \mid 2 \leq i \land \text{real } i \leq \log 2 x) \vartheta (\text{root } i x)\) using finite
  by (subst sum.insert) auto
finally show \(?thesis\).
qed

The following theorems use summation by parts to relate different prime-counting functions to one another with an integral as a remainder term.

lemma \(\vartheta\)-conv-\(\pi\)-integral:
assumes \(x: x \geq 2\)
shows \(((\lambda t. \pi t / t) \text{ has-integral} (\pi x * \ln x - \vartheta x)) \{2..x\}\)
proof (cases \(x = 2\))
case False
note [intro] = \(\text{finite-vimage-real-of-nat-greaterThanAtMost}\)
from False and assms have \(x: x > 2\) by simp
have \(((\lambda t. \text{sum-upto} (\text{ind prime}) t * (1 / t)) \text{ has-integral}
\text{sum-upto} (\text{ind prime}) x * \ln x - \text{sum-upto} (\text{ind prime}) 2 * \ln 2 -
(\sum n \in\text{real } \rightarrow \{2..x\}. \text{ind prime } n * \ln (\text{real } n))\) \(\{2..x\}\) using \(x\)
  by (intro partial-summation-strong[where \(X = \{\}\)])
(auto intro!; continuous-intros derivative-eq-intros
  simp flip: has-field-derivative-iff-has-vector-derivative)
hence \(((\lambda t. \pi t / t) \text{ has-integral} (\pi x * \ln x -
(\pi 2 * \ln 2 + (\sum n \in\text{real } \rightarrow \{2..x\}. \text{ind prime } n * \ln n))\) \(\{2..x\}\)
  by (simp add: \(\pi\)-def prime-sum-upto-altdef1 algebra-simps)
also have \(\pi 2 * \ln 2 + (\sum n \in\text{insert } 2 (\text{real } \rightarrow \{2..x\}). \text{ind prime } n * \ln n) =
(\sum n \in\text{insert } 2 (\text{real } \rightarrow \{2..x\}). \text{ind prime } n * \ln n)\)
  by (subst sum.insert) (auto simp: eval-\(\pi\))
also have \(\ldots = \vartheta x\) unfolding \(\vartheta\)-def prime-sum-upto-def using \(x\)
by (intro sum.mono-neutral-cong-right) (auto simp: ind-def dest: prime-gt-1-nat)
finally show \(?thesis\).
qed (auto simp: has-integral-refl eval-\(\pi\) eval-\(\vartheta\))

lemma \(\pi\)-conv-\(\vartheta\)-integral:
assumes \(x: x \geq 2\)
shows \(((\lambda t. \vartheta t / (t * \ln t \land 2)) \text{ has-integral} (\pi x - \vartheta x / \ln x)) \{2..x\}\)
proof (cases \(x = 2\))

case False

define \(b\) where \(b = (\lambda p. \text{ind prime } p \ast \ln (\text{real } p))\)

note [intro] = finite-vimage-real-of-nat-greaterThanAtMost

from False and assms have \(x : x > 2\) by simp

have \(((\lambda t. - (\text{sum-upto } b \ast \ln \ X \ t) \ t * (-1 / (t * (\ln \ t) ^ 2)))\) has-integral

\(- (\text{sum-upto } b \ast \ln \ X \ t) \ t * (1 / (\ln \ t)) - \text{sum-upto } b \ast \ln \ X \ t * (1 / (\ln \ 2)) - \)

\((\sum n \in \text{real} \ - \ (2 <..x). \ b \ast (1 / (\ln \ (\text{real } n)))) \ (2..x)\) using \(x\)

by (intro has-integral-neg partial-summation-strong[where \(X = \{\}]\)

(auto intro!: continuous-intros derivative-eq-intros

simp flip: has-field-derivative-iff-has-vector-derivative simp add: power2-eq-square)

also have \(\text{sum-upto } b = \vartheta\)

by (simp add: \(\vartheta\)-def b-def prime-up-to-altdef1 fun-eq-iff)

also have \(\vartheta x * (1 / (\ln \ X \ t)) - \vartheta 2 * (1 / (\ln \ 2)) - \)

\((\sum n \in \text{real} \ - \ (2 <..x). \ b \ast (1 / (\ln \ (\text{real } n)))) = \)

\(\vartheta x * (1 / (\ln \ X \ t)) - (\sum n \in \text{insert} \ (\text{real} \ - \ (2 <..x)). \ b \ast (1 / (\ln \ (\text{real } n)))) = \pi x\) using \(x\)

unfolding \(\pi\)-def prime-up-to-altdef1 sum-up-to-def

proof (intro sum.mono-neutral-cong-left ballII, goal-cases)

case (3 \(p\))

hence \(p = 1\) by auto

thus \(\bar{\epsilon}\) case by auto

qed (auto simp: b-def)

finally show \(?\)thesis by simp

qed (auto simp: has-integral-refl eval-\(\pi\) eval-\(\vartheta\))

glemma integrable-weighted-\(\vartheta\):

assumes \(2 \leq a \leq x\)

shows \(((\lambda t. \vartheta t / (t * \ln \ t \ (-2)))\) integrable-on \(\{a..x\}\)

proof (cases \(a < x\))

case True

hence \(((\lambda t. \vartheta t / (t * \ln \ t \ (-2)))\) integrable-on \(\{a..x\}\)) using assms

unfolding \(\vartheta\)-def prime-up-to-altdef1

by (intro partial-summation-integrable-strong[where \(X = \{\}]\) and \(f = \lambda x. -1 / (\ln \ X)\)

(auto simp flip: has-field-derivative-iff-has-vector-derivative

intro!: derivative-eq-intros continuous-intros simp: power2-eq-square

field-simps)

thus \(?\)thesis by simp

qed (insert has-integral-refl[of - \(a\)] assms, auto simp: has-integral-iff)

glemma \(\vartheta\)-cone-\(\mathfrak{M}\)-integral:

assumes \(x \geq \bar{\epsilon}\)

shows \(\{(\mathfrak{M} \ has-integral) (\mathfrak{M} \ x * x - \vartheta x)\} \ (2..x)\)

proof (cases \(x = 2\))

case False

72
with assms have \( x \colon x > 2 \) by simp

define \( b \colon \text{nat} \Rightarrow \text{real} \) where \( b = (\lambda p. \text{ind prime p * ln p / p}) \)

note \([\text{intro}]= \text{finite-vimage-real-of-nat-greaterThanAtMost} \)

have \( \text{prime-le-2} \colon p = 2 \) if \( p \leq 2 \) prime \( p \) for \( p \colon \text{nat} \)
  using that by (auto simp: prime-nat-iff)

have \( ((\lambda t. \text{sum-upto } b \; t \cdot 1)) \text{ has-integral} \text{ sum-upto } b \; x \cdot x - \text{sum-upto } b \; 2 \cdot 2 \)
  
  \( (\sum n \in \text{real} -' \{2<x\}, \; b \; n \cdot \text{real } n) \) \{2..x\} using \( x \)
  by (intro partial-summation-strong[of \{}\])
  (auto simp flip: has-field-derivative-iff-has-vector-derivative
    intro!: derivative-eq-intros continuous-intros)

also have \( \text{sum-upto } b = M \) by (simp add: fun-eq-iff primes-M-def b-def prime-sum-upto-altdef1)

also have \( M \cdot x \cdot x - M \cdot 2 \cdot 2 - (\sum n \in \text{real} -' \{2<x\}, \; b \cdot n \cdot \text{real } n) = \)

  \( \text{sum-upto } \{2..x\}, \; b \cdot n \cdot \text{real } n\) using \( x \)

unfolding \( \vartheta\)-def prime-sum-upto-def using \( x \)

by (intro sum-mono-neutral-cong-right) (auto simp: b-def ind-def not-less prime-le-2)

finally show \( ?\text{thesis} \) by simp

qed (auto simp: eval-\( \vartheta \) eval-M)

lemma \( \text{integrable-primes-M} \colon \mathbb{R} \text{ integrable-on} \{x..y\} \) if \( 2 \leq x \) for \( x \; y \colon \text{real} \)

proof –

have \( (\lambda x. \mathbb{R} \; x \cdot 1) \text{ integrable-on} \{x..y\} \) if \( 2 \leq x < y \) for \( x \; y \colon \text{real} \)

unfolding primes-M-def prime-sum-upto-altdef1 using that

by (intro partial-summation-integrable-strong[where \( X = \{\} \) and \( f = \lambda x. \; x\)])

(auto simp flip: has-field-derivative-iff-has-vector-derivative
  intro!: derivative-eq-intros continuous-intros)

thus \( ?\text{thesis} \) using that has-integral-refl(2)[of \( \mathbb{R} \; x\)] by (cases x y rule: linorder-cases)

auto

qed

3.5 Bounds

lemma \( \vartheta\)-upper-bound-coarse:

assumes \( x \geq 1 \)

shows \( \vartheta \cdot x \leq x \cdot \ln x \)

proof –

have \( \vartheta \cdot x \leq \text{sum-upto } (\lambda -. \ln x) \) \( \text{unfolding } \vartheta\)-def prime-sum-upto-altdef1

sum-upto-def

by (intro sum-mono) (auto simp: ind-def)

also have \( \ldots \leq \text{real-of-int } |x| \cdot \ln x \) using assms

by (simp add: sum-upto-altdef)

also have \( \ldots \leq x \cdot \ln x \) using assms by (intro mult-right-mono) auto

finally show \( ?\text{thesis} \).

qed
lemma \( \vartheta \text{-le-}\psi \): \( \vartheta \ x \leq \psi \ x \)
proof (cases \( x \geq 2 \))
  case False
  hence \( \text{nat} \ [x] = 0 \lor \text{nat} \ [x] = 1 \) by linarith
  thus \( \exists \text{thesis} \) by (auto simp: eval-\( \vartheta \))
next
case True
  hence \( \psi \ x - \vartheta \ x = (\sum i | \ 2 \leq i \land \text{real} \ i \leq \log 2 \ x \cdot \vartheta \ (\text{root} \ i \ x)) \)
  by (rule \( \psi \text{-minus-}\vartheta \))
  also have \( \ldots \geq 0 \) by (intro sum-nonneg) auto
  finally show \( \exists \text{thesis} \) by simp
qed

lemma \( \pi \text{-upper-bound-coarse} \): 
assumes \( x \geq 0 \)
shows \( \pi \ x \leq x / 3 + 2 \)
proof
  have \( \{ p. \ prime \ p \land \ p \leq \text{natt} \ [x] \} \subseteq \{2, 3\} \cup \{p. \ p \neq 1 \land \text{odd} \ p \land \neg 3 \text{ dvd} \ p \land \ p \leq \text{natt} \ [x]\} \)
  using primes-dvd-imp-eq[of 2 :: \text{natt}] primes-dvd-imp-eq[of 3 :: \text{natt}] by auto
  also have \( \ldots \subseteq \{2, 3\} \cup (\lambda k. 6k+1) \cdot \{0 \leq \ldots \text{natt} \ [(x+5)/6]\} \cup (\lambda k. 6k+5) \cdot \{..\text{natt} \ [(x+1)/6]\}\) 
  (is \( \subseteq \) ?lhs <= - Union ?rhs)
  proof (intro Un-mono subsetI)
    fix \( p :: \text{natt} \)
    assume \( p \in \ ?lhs \)
    hence \( p \neq 1 \land \text{odd} \ p \land 3 \text{ dvd} \ p \land \ p \leq \text{natt} \ [x] \) by auto
    from \( p \ (1-3) \) have \( \exists k. \ k > 0 \land p = 6 \cdot k + 1 \lor p = 6 \cdot k + 5 \) by presburger
    then obtain \( k \) where \( k > 0 \land p = 6 \cdot k + 1 \lor p = 6 \cdot k + 5 \) by blast
    hence \( p = 6 \cdot k + 1 \lor k > 0 \land k < \text{natt} \ [(x+5)/6] \lor p = 6\cdot k+5 \land k < \text{natt} \ [(x+1)/6] \)
    unfolding add-divide-distrib using \( p(4) \) by linarith
    thus \( p \in \ ?rhs \) by auto
  qed
  finally have subset: \( \{ p. \ prime \ p \land \ p \leq \text{natt} \ [x] \} \subseteq \ldots \ (\is \subseteq \ ) \).
  have \( \pi \ x \leq \text{real} \ (\text{card} \ \{ p. \ prime \ p \land \ p \leq \text{natt} \ [x] \}) \)
  by (simp add: \( \pi \text{-def} \ \text{prime-sum-upto-aldef2} \))
  also have \( \text{card} \ \{ p. \ prime \ p \land \ p \leq \text{natt} \ [x] \} \leq \text{card} \ ?A \)
  by (intro card-mono subset) auto
  also have \( \ldots \leq 2 + (\text{natt} \ [(x+5)/6] - 1 + \text{natt} \ [(x+1)/6]) \)
  by (intro order:trans[OF card-Un-le] add-mono order:trans[OF card-image-le])
  auto
  also have \( \ldots \leq x / 3 + 2 \)
  using \( \text{assms} \) unfolding add-divide-distrib by (cases \( x \geq 1 \), linarith, simp)
  finally show \( \exists \text{thesis} \) by simp
qed

lemma \( \text{le-numeral-iff} \): \( m \leq \text{numeral} \ n \iff m = \text{numeral} \ n \lor m \leq \text{pred-numeral} \)

74
The following nice proof for the upper bound $\theta(x) \leq \ln 4 \cdot x$ is taken from Otto Forster’s lecture notes on Analytic Number Theory [4].

**Lemma** $\prod$-primes-upto-less:
- **defines** $F \equiv (\lambda n. (\prod \{ p\::\text{nat}. \text{prime } p \land p \leq n\} ))$
- **shows** $n > 0 \implies F n < 4 ^ n$

**Proof** (induction $n$ rule: less-induct)
- **case** $(\text{less } n)$
  - **have** $n = 0 \lor n = 1 \lor n = 2 \lor n = 3 \lor \text{even } n \land n \geq 4 \lor \text{odd } n \land n \geq 4$
    - **by** presburger
  - **then consider** $n = 0 \mid n = 1 \mid n = 2 \mid n = 3 \mid \text{even } n \ n \geq 4 \mid \text{odd } n \ n \geq 4$
    - **by** metis
  - **thus** @case

**Proof** cases
- **assume** [@simp]; $n = 1$
  - **have** $\{ p. \text{prime } p \land p \leq \text{Suc } 0\} = \{\}$ **by** (auto dest: prime-gt-1-nat)
  - **show** ?thesis **by** (simp add: $F$-def @*)

next
- **assume** [@simp]; $n = 2$
  - **have** $\{ p. \text{prime } p \land p \leq 2\} = \{2 :: \text{nat}\}$
    - **by** (auto simp: le-numeral-iff dest: prime-gt-1-nat)
  - **thus** ?thesis **by** (simp add: $F$-def @*)

next
- **assume** [@simp]; $n = 3$
  - **have** $\{ p. \text{prime } p \land p \leq 3\} = \{2, 3 :: \text{nat}\}$
    - **by** (auto simp: le-numeral-iff dest: prime-gt-1-nat)
  - **thus** ?thesis **by** (simp add: $F$-def @*)

next
- **assume** $n$: even $n \ n \geq 4$
  - **from** $n$ **have** $F (n - 1) < 4 ^ (n - 1)$ **by** (intro less.IH) auto
  - **also** have $\text{prime } p \land p \leq n \iff \text{prime } p \land p \leq n - 1$ **for** $p$
    - **using** $\text{prime-odd-nat[of } n\}$ **by** (cases $p = n$) auto
  - **hence** $F (n - 1) = F n$ **by** (simp add: $F$-def)
  - **also** have $4 ^ (n - 1) \leq 4 ^ n :: \text{nat}$ **by** (intro power-increasing) auto
  - **finally** show ?case .

next
- **assume** $n$: odd $n \ n \geq 4$
  - **then obtain** $k$ **where** $k$-eq: $n = \text{Suc } (2 * k)$ **by** (auto elim: oddE)
  - **from** $n$ **have** $k$: $k \geq 2$ **unfolding** $k$-eq **by** presburger
    - **have** $\text{prime-dvd: } p \ \text{dvd} (n \ \text{choose } k)$ **if** $p$: prime $p \ p \in \{k+1..<n\}$ **for** $p$
      - **proof**
        - **from** $p \ k \ n$ **have** $p \ \text{dvd} \ \text{pochhammer} (k + 2) \ k$
          - **unfolding** pochhammer-prod
            - **by** (subst prime-dvd-prod-iff)
              - (auto intro!: bexI[of - $p - k - 2$] simp: $k$-eq numeral-2-eq-2 Suc-diff-Suc)
            - **also** have $\text{pochhammer} \ (\text{real } (k + 2)) \ k = \text{real } (\{n \ \text{choose } k\} * \text{fact } k)$
              - **by** (simp add: binomial-gbinomial gbinomial-pochhammer' $k$-eq field-simps)
have $\prod \{ p. \ prime \ p \land \ p \in \{ k+1<..n \}\} \ dvd \ (n \ choose \ k)$

thus $\exists \ case$

proof (rule multiplicity-le-imp-dvd, goal-cases)
  case (2 \ p)
  by (auto simp: prime-dvd-fact-iff prime-dvd-mult-nat)

qed

also have $\vdots = \sum (multiplicity \ p) \ {\{ p \}}$ using True 2

by (cases \ p = {q}) (auto simp: prime-multiplicity-other)

qed auto

also have $\vdots = 1$ using 2 by simp

also have $1 \leq \ multiplicity \ p \ (n \ choose \ k)$

using prime-dvd[of \ p] 2 True by (intro multiplicity-geI) auto

finally show $\vdots$ thesis .

qed auto

hence $\prod \{ p. \ prime \ p \land \ p \in \{ k+1<..n \}\} \leq (n \ choose \ k)$

by (intro dvd-imp-le) (auto simp: k-eq)

also have $\vdots = 1 / 2 * (\sum \ i\in\{ k, \ Suc \ k \}\, n \ choose \ i)$

using central-binomial-odd[of \ n] by (simp add: k-eq)

also have $(\sum \ i\in\{ k, \ Suc \ k \}\, n \ choose \ i) < (\sum \ i\in\{ 0, \ k, \ Suc \ k \}\, n \ choose \ i)$

using k by simp

also have $\vdots = (1 + 1) ^ n$

by (intro sum-mono2) (auto simp: k-eq)

also have $\vdots = (1 + 1) ^ n$

using binomial[of \ 1 \ n] by simp

also have $1 / 2 * \vdots = real (4 ^ k)$

by (simp add: k-eq power-mult)

finally have less: $(\prod \{ p. \ prime \ p \land \ p \in \{ k+1<..n \}\}) < 4 ^ k$

unfolding of-nat-less-iff by simp

have $F \ n = F \ (Suc \ k) * (\prod \{ p. \ prime \ p \land \ p \in \{ k+1<..n \}\})$ unfolding F-def

by (subst prod.union-disjoint [symmetric]) (auto intro!: prod.cong simp: k-eq)
also have \( \ldots < 4 \cdot Suc \ k \cdot 4 \cdot k \) using \( n \)
by \((\text{intro mult-strict-mono less.IH})\) \((\text{auto simp: k-eq})\)
also have \( \ldots = 4 \cdot (Suc \ k + k) \)
by \((\text{simp add: power-add})\)
also have \( Suc \ k + k = n \) by \((\text{simp add: k-eq})\)
finally show \( ?\text{case} \).
qed \((\text{insert less.prems, auto})\)

\[ \text{lemma } \vartheta\text{-upper-bound:} \]
assumes \( x: x \geq 1 \)
shows \( \vartheta \cdot x < \ln 4 \cdot x \)
proof –
have \( 4 \powr (\vartheta \cdot x / \ln 4) = (\prod p | \text{prime } p \land p \leq \text{nat } [x]. \ 4 \powr (\log 4 (\text{real } p))) \)
by \((\text{simp add: \( \vartheta\)-def powr-sum prime-sum-upto-altdef2 sum-divide-distrib log-def})\)
also have \( \ldots = (\prod p | \text{prime } p \land p \leq \text{nat } [x]. \ \text{real } p) \)
by \((\text{intro prod.cong})\) \((\text{auto dest: prime-gt-1-nat})\)
also have \( \ldots = \text{real } (\prod p | \text{prime } p \land p \leq \text{nat } [x]. \ p) \)
by \(\text{simp}\)
also have \( (\prod p | \text{prime } p \land p \leq \text{nat } [x]. \ p) < 4 \cdot \text{nat } [x] \)
using \( x \)
by \((\text{intro prod-primes-upto-less})\) \(\text{auto}\)
also have \( \ldots \leq 4 \powr x \)
using \( x \)
by \((\text{intro powr-mono})\) \(\text{auto}\)
finally have \( 4 \powr (\vartheta \cdot x / \ln 4) < 4 \powr x \)
by \(\text{simp}\)
thus \( \vartheta \cdot x < \ln 4 \cdot x \)
by \((\text{subst (asm) powr-less-cancel-iff})\) \((\text{auto simp: field-simps})\)
qed

\[ \text{lemma } \vartheta\text{-bigo: } \vartheta \in O(\lambda x. \ x) \]
by \((\text{intro le-imp-bigo-real[of ln 4] eventually-mono[of OF eventually-ge-at-top[of 1]] less-imp-le[of \( \vartheta\)-upper-bound])\) \(\text{auto}\)

\[ \text{lemma } \psi\text{-minus-\( \vartheta\)-bound:} \]
assumes \( x: x \geq 2 \)
shows \( \psi \cdot x - \vartheta \cdot x \leq 2 \cdot \ln x \cdot \sqrt x \)
proof –
have \( \psi \cdot x - \vartheta \cdot x = (\sum i | 2 \leq i \land \text{real } i \leq \log 2 x. \ \vartheta \cdot (\text{root } i \cdot x)) \) using \( x \)
by \((\text{rule \( \psi\)-minus-d})\)
also have \( \ldots \leq (\sum i | 2 \leq i \land \text{real } i \leq \log 2 x. \ ln 4 \cdot \text{root } i \cdot x) \)
using \( x \)
by \((\text{intro sum-mono less-imp-le[of \( \vartheta\)-upper-bound])\) \(\text{auto}\)
also have \( \ldots \leq (\sum i | 2 \leq i \land \text{real } i \leq \log 2 x. \ ln 4 \cdot \text{root } 2 \cdot x) \) using \( x \)
by \((\text{intro sum-mono mult-mono})\) \((\text{auto simp: le-log-iff powr-realpow intro!: real-root-decreasing})\)
also have \( \ldots = \text{card } \{i. \ 2 \leq i \land \text{real } i \leq \log 2 x\} \cdot \ln 4 \cdot \sqrt x \)
by \((\text{simp add: sqrt-def})\)

77
also have \{ i. \ 2 \leq i \land \text{real } i \leq \log 2 x \} = \{ \text{2..nat } \lfloor \log 2 x \rfloor \}
  by (auto simp: le-nat-iff le-floor-iff)
also have \log 2 x \geq 1 using \ x by (simp add: le-log-iff)
hence real (\lfloor \log 2 x \rfloor - 1) \leq \log 2 x using \ x by linarith
hence \log 2 x \ast \ln 4 \ast \sqrt x = 2 \ast \ln x \ast \sqrt x using \ x
finally show \ ?thesis using \ x by (simp add: mult-right-mono)

We shall now attempt to get some more concrete bounds on the difference between \( \pi(x) \) and \( \theta(x)/\ln x \). These will be essential in showing the Prime Number Theorem later.

We first need some bounds on the integral
\[ \int_2^x \frac{1}{\ln^2 t} \, dt \]
in order to bound the contribution of the remainder term. This integral actually has an antiderivative in terms of the logarithmic integral \( \text{li}(x) \), but since we do not have a formalisation of it in Isabelle, we will instead use the following ad-hoc bound given by Apostol:

**Lemma** integral-one-over-log-squared-bound:
- assumes \( x: x \geq 4 \)
- shows \( \int_2^x \frac{1}{\ln^2 t} \, dt \leq \frac{\sqrt x}{\ln 2} + \frac{4 * x}{\ln x} \)

**Proof**
- from \( x \) have \( x * 1 \leq x \ast 2 \) unfolding power2-eq-square by (intro mult-left-mono)
- with \( x \) have \( \sqrt x \ast \sqrt x \leq x \)
- by (auto simp: real-sqrt-le-iff intro: real-le-rsqrt)
\begin{verbatim}

have integral {2..x} (λt. 1 / ln t ^ 2) =
  integral {2..sqrt x} (λt. 1 / ln t ^ 2) + integral {sqrt x..x} (λt. 1 / ln t ^ 2)
(\text{is } = \text{?I1 + ?I2) using } x'
\text{by (intro integral-combine [symmetric] integrable-continuous-real)}
\text{(auto intro!: continuous-intros)}
\text{also have } ?I1 ≤ integral \{2..sqrt x\} (λt. 1 / ln 2 ^ 2) using x
\text{by (intro integral-le integrable-continuous-real divide-left-mono power-mono continuous-intros) auto}
\text{also have } \ldots ≤ sqrt x / ln 2 ^ 2 using x' by (simp add: field-simps)
\text{also have } ?I2 ≤ integral {sqrt x..x} (λt. 1 / ln (sqrt x) ^ 2) using x'
\text{by (intro integral-combine integrable-continuous-real divide-left-mono power-mono continuous-intros) auto}
\text{also have } \ldots ≤ 4 * x / ln x ^ 2 using x' by (simp add: ln-sqrt field-simps)
finally show ?thesis by simp

qed

lemma integral-one-over-log-squared-bigo:
(λx::real. integral \{2..x\} (λt. 1 / ln t ^ 2)) ∈ O(λx. x / ln x ^ 2)
\text{proof –}
\text{define ub where } ub = \text{(λx::real. sqrt x / ln 2 ^ 2 + 4 * x / ln x ^ 2)}
\text{have eventually (λx. integral \{2..x\} (λt. 1 / (ln t)^2) \leq |ub x| at-top)}
\text{using eventually-ge-at-top[of 4]}
\text{proof eventually-elim}
\text{case (elim x)}
\text{hence |integral \{2..x\} (λt. 1 / ln t ^ 2)| = integral \{2..x\} (λt. 1 / ln t ^ 2)}
\text{by (intro abs-of-nonneg integral-nonneg integrable-continuous-real continuous-intros)}
\text{auto}
\text{also have } \ldots ≤ |ub x|
\text{using integral-one-over-log-squared-bound[of x] elim by (simp add: ub-def)}
finally show ?case .

qed

hence (λx. integral \{2..x\} (λt. 1 / (ln t)^2)) ∈ O(ub)
\text{by (intro landau-o.bigI[of 1]) auto}
\text{also have } ub ∈ O(λx. x / ln x ^ 2) unfolding ub-def by real-asymp
finally show ?thesis .

qed

lemma π-ρ-bound;
\text{assumes } x ≥ (4 :: real)
\text{defines } ub ≡ 2 / ln 2 * sqrt x + 8 * ln 2 * x / ln x ^ 2
\text{shows } π * x - ρ * x / ln x ∈ \{0..ub\}
\text{proof –}
\text{define r where } r = \text{(λx. integral \{2..x\} (λt. ρ t / (t + ln t ^ 2))}
\text{have integrable: (λt. c / ln t ^ 2) integrable-on \{2..x\} for c}
\text{by (intro integrable-continuous-real continuous-intros) auto}
\text{have r x ≤ integral \{2..x\} (λt. ln 4 / ln t ^ 2) unfolding r-def}
\text{using integrable-weighted-ρ[of 2 x] integrable[of ln 4] asms less-imp-le[OF}

\end{verbatim}
\[ \vartheta \text{-upper-bound} \]

by (intro integral-le divide-right-mono) (auto simp: field-simps)
also have \ldots \leq \ln \frac{4}{x} * \text{integral} \{2..x\} (\lambda t. 1 / \ln t \cdot 2)
using integrable[of 1] by (subst integral-mult) auto
also have \ldots \leq \ln \frac{4}{x} (\sqrt{x} / \ln 2 \cdot 2 + 4 * x / \ln x \cdot 2)
using assms by (intro mult-left-mono integral-one-over-log-squared-bound) auto
also have \ln (\frac{4}{x} : \text{real}) = 2 * \ln 2
using ln-realpow[of 2] 2 by simp
also have \ldots \ast (\sqrt{x} / \ln 2 \cdot 2 + 4 * x / \ln x \cdot 2) = \ub
using assms by (simp add: field-simps power2-eq-square ub-def)
finally have \r x \leq \ldots .
moreover have \r x \geq 0 unfolding r-def using assms
by (intro integral-nonneg integrable-weighted-\vartheta divide-nonneg-pos) auto
ultimately have \r x \in \{0..\ub\} by auto
with \pi-conv-\vartheta-integral[of x] \assms(1) show \thesis
by (simp add: r-def has-integral-iff)
qed

The following statement already indicates that the asymptotics of \pi and \vartheta are very closely related, since through it, \pi(x) \sim x / \ln x and \vartheta(x) \sim x imply each other each other.

\textbf{lemma} \pi-\vartheta-bigo: \(\lambda x. \pi x - \vartheta x / \ln x \in O(\lambda x. x / \ln x \cdot 2)\)

\textbf{proof} –
define \ub where \ub = \(\lambda x. 2 / \ln 2 \ast \sqrt{x} + 8 \ast \ln 2 \ast x / \ln x \cdot 2\)
have \(\lambda x. \pi x - \vartheta x / \ln x \in O(\ub)\)
proof (intro le-imp-bigo-real[of 1] eventually-mono[of \ub eventually-ge-at-top])
fix x :: real assume \x \geq 4
from \pi-\vartheta-bound[OF this] show \pi x - \vartheta x / \ln x \geq 0 and \pi x - \vartheta x / \ln x \leq 1 * \ub x
by (simp-all add: ub-def)
qed auto
also have \ub \in O(\lambda x. x / \ln x \cdot 2)
unfolding ub-def by real-asym
finally show \thesis .
qed

As a foreshadowing of the Prime Number Theorem, we can already show the following upper bound on \pi(x):

\textbf{lemma} \pi-upper-bound:

assumes \x \geq (4 :: real)
shows \pi x < \ln 4 * x / \ln x + 8 * \ln 2 * x / \ln x \cdot 2 + 2 / \ln 2 * \sqrt{x}

\textbf{proof} –
define \ub where \ub = 2 / \ln 2 * \sqrt{x} + 8 * \ln 2 * x / \ln x \cdot 2
have \pi x \leq \vartheta x / \ln x + \ub
using \pi-\vartheta-bound[of x] assms unfolding ub-def by simp
also from assms have \vartheta x / \ln x < \ln \frac{4}{x} * x / \ln x
by (intro \vartheta-upper-bound divide-strict-right-mono) auto
finally show \thesis
using assms by (simp add: algebra-simps ub-def)
lemma π-bigo: π ∈ O(λx. x / ln x)
proof
have (λx. π x − ϑ x / ln x) ∈ O(λx. x / ln x) by (fact π-ϑ-bigo)
also have (λx::real. x / ln x^2) ∈ O(λx. x / ln x) by real-asym
finally have (λx. π x − ϑ x / ln x) ∈ O(λx. x / ln x) by (rule sum-in-bigo)
thus thesis by simp
qed

3.6 The asymptotic form of Mertens’ First Theorem

Mertens’ first theorem states that \( M(x) - \ln x \) is bounded, i.e. \( M(x) = \ln x + O(1) \).
With some work, one can also show some absolute bounds for \( |M(x) - \ln x| \), and we will, in fact, do this later. However, this asymptotic form is somewhat easier to obtain and it is (as we shall see) enough to prove the Prime Number Theorem, so we prove the weak form here first for the sake of a smoother presentation.

First of all, we need a very weak version of Stirling’s formula for the logarithm of the factorial, namely:

\[
\ln(\lfloor x! \rfloor) = \sum_{n \leq x} \ln n = x \ln x + O(x)
\]

We show this using summation by parts.

lemma stirling-weak:
assumes x: x ≥ 1
shows sum-upto ln x ∈ {x * ln x − x − ln x + I .. x * ln x}
proof (cases x = 1)
case True
have {0<..<Suc 0} = {1} by auto
with True show thesis by (simp add: sum-upto-altdef)
ext
next case False
with assms have x: x > 1 by simp
have (\( \lambda t. \sum_{x \in \text{real} - \{1<..<x\}} t \star (1 / t) \)) has-integral
sum-upto (λx. t * ln x - sum-upto (λx. 1 * ln 1 - \( \sum_{n \in \text{real} - \{1<..<x\}} 1 \star (\text{real} n) \)) {1..x}) using x
by (intro partial-summation-strong[of {}])
(auto simp flip: has-field-derivative-iff-has-vector-derivative
intro!: derivative-eq-intros continuous-intros)

hence \(((\lambda. \text{real } (\text{nat } t))/t)\) has-integral
real (\text{nat } x) * ln x - (∑ n∈real - 1<..x). ln (\text{real } n)) \{1..x\}
by (simp add: sum-upto-altdef)
also have (∑ n∈real - 1<..x). ln (real n)) = sum-upto ln x
unfolding sum-upto-def
by (intro sum.mono-neutral-left)
(auto intro!: finite-subset[OF finite-vimage-real-of-nat-greaterThanAtMost[of 0 x]])
finally have \(*: ((\lambda. \text{real } (\text{nat } t))/t)\) has-integral \([x] * \ln x - \text{sum-upto ln x}\) \{1..x\}
using x by simp

have 0 ≤ real-of-int \([x] * \ln x - \text{sum-upto } (\lambda n. \ln (\text{real } n))\) \(x\)
using * by (rule has-integral-nonneg) auto
also have \(\ldots \leq x * \ln x - \text{sum-upto ln x}\)
using x by (intro diff-mono mult-mono) auto
finally have upper: sum-upto ln x \(\leq x * \ln x\) by simp

have \((x - 1) * \ln x - x + 1 \leq \lfloor x \rfloor * \ln x - x + 1\)
using x by (intro diff-mono mult-mono add-mono) auto
also have \((\lambda. 1)\) has-integral \((x - 1)\) \{1..x\}
using has-integral-const-real[of 1::real 1 x] \(x\) by simp
from * and this have \([x] * \ln x - \text{sum-upto ln x} \leq x - 1\)
by (rule has-integral-le) auto
hence \([x] * \ln x - x + 1 \leq \text{sum-upto ln x}\)
by simp
finally have sum-upto ln x \(\geq x * \ln x - \ln x + 1\)
by (simp add: algebra-simps)
with upper show ?thesis by simp
qed

lemma stirling-weak-bigo: \((\lambda x::\text{real}. \text{sum-upto } \ln x - x * \ln x)\) ∈ \(O(\lambda x. \ x)\)
proof –
have \((\lambda x. \text{sum-upto } \ln x - x * \ln x)\) ∈ \(O(\lambda x. - (\text{sum-upto } \ln x - x * \ln x))\)
by (subst landau-o.big.unminus) auto
also have \((\lambda x. -(\text{sum-upto } \ln x - x * \ln x))\) ∈ \(O(\lambda x. \ x + \ln x - 1)\)
proof (intro le-imp-bigo-real[of 2] eventually-mono[OF eventually-ge-at-top[of 1]], goal-cases)
case (2 x)
thus ?case using stirling-weak[of x] by (auto simp: algebra-simps)
next
case (3 x)
thus ?case using stirling-weak[of x] by (auto simp: algebra-simps)
qed auto
also have \((\lambda x. \ x + \ln x - 1)\) ∈ \(O(\lambda x::\text{real}. \ x)\) by real-asym
finally show ?thesis .
qed
The key to showing Mertens’ first theorem is the function
\[ h(x) := \sum_{n \leq x} \frac{\Lambda(d)}{d} \]

where \( \Lambda \) is the Mangoldt function, which is equal to \( \ln p \) for any prime power \( p^k \) and 0 otherwise. As we shall see, \( h(x) \) is a good approximation for \( \mathfrak{M}(x) \), as the difference between them is bounded by a constant.

**Lemma**: sum-upto-mangoldt-over-id-minus-phi-bounded:
\[
(\lambda x. \text{sum-upto} (\lambda d. \text{mangoldt} d / \text{real} d) x - \mathfrak{M} x) \in O(\lambda^- 1)
\]

**Proof** –
- **Define** \( f \) where \( f = (\lambda d. \text{mangoldt} d / \text{real} d) \)
- **Define** \( C \) where \( C = (\sum p. \ln (\text{real } (p + 1)) \times (1 / \text{real } (p * (p - 1)))) \)
- **Have** summable: summable \((\lambda p::\text{nat}. \ln (p + 1) \times (1 / (p * (p - 1))))\)
- **Proof** (rule summable-comparison-test-bigo)
  - **Show** summable \((\lambda p. \text{norm } (p \text{ powr } (-3/2)))\)
  - **By** (simp add: summable-real-powr-iff)

**QED** real-asympt

**Have** diff-bound: sum-upto \( f \) \( x - \mathfrak{M} x \in \{0..C \} \) if \( x: x \geq 4 \) for \( x \)

**Proof** –
- **Define** \( S \) where \( S = \{ (p, i). \text{prime } p \land 0 < i \land \text{real } (p ^{\ i} i) \leq x \} \)
- **Define** \( S' \) where \( S' = (\Sigma p:2..\text{nat} \{ \text{root } 2 \ x \} ) \times \{2..\text{nat} \{ \text{log } 2 \ x \} \} \)
- **Have** \( S \subseteq \{ \text{nat } \{ x \} \} \times \{ \text{nat } \{ \text{log } 2 \ x \} \} \) unfolding \( S\)-def
  - **Using** \( x \text{ primepows-le-subset[of } x I \text{] by } (\text{auto simp: Suc-le-eq}) \)
  - **Hence** finite \( S \) by (rule finite-subset) auto
  - **Note** fin = finite-subset[OF - this, unfolded \( S\)-def]

**Have** sum-upto \( f \) \( x = (\sum (p, i) \in S. \ln (\text{real } p) / \text{real } (p ^{\ i} i)) \) unfolding \( S\)-def
  - **By** (intro sum-upto-primepows) (auto simp: f-def mangoldt-non-primepow)
  - **Also have** \( S = \{ p. \text{prime } p \land p \leq x \} \times \{1\} \cup \{(p, i). \text{prime } p \land 1 < i \land \text{real } (p ^{\ i} i) \leq x \} \)
    - **By** (auto simp: S-def not-less le-Suc-eq not-le intro! Suc-lessI)
    - **Also have** \( \sum (p, i) \in \ldots \ln (\text{real } p) / \text{real } (p ^{\ i} i) = \)
      \( (\sum (p, i) \in \{ \text{prime } p \land p \leq x \} \times \{1\} \ln (\text{real } p) / \text{real } (p ^{\ i} i) + \)
      \( \sum (p, i) | \text{prime } p \land \text{real } (p ^{\ i} i) \leq x \land i > 1. \ln (\text{real } p) / \text{real } (p ^{\ i} i) \)
    - (is - = \( ?S1 + ?S2 \))
Next, we show that our $h(x)$ itself is close to $\ln x$, i.e.:

$$\sum_{n\le x} \frac{A(d)}{d} = \ln x + O(1)$$
lemma sum-upto-mangoldt-over-id-asymptotics:
(λx. sum-upto (λd. mangoldt d / real d) x − ln x) ∈ O(λx. 1)

proof

define r where r = (λn::real. sum-upto (λd. mangoldt d * (n / d − real-of-int [n / d])) n)

have r: r ∈ O(ψ)

proof (intro landau-o.bigi[of 1] eventually-mono[OF eventually-ge-at-top[of 0]])

fix x :: real assume x: x ≥ 0

have eq: {1..<nat [x]} = {0..<nat [x]} by auto

hence r x ≥ 0 unfolding r-def sum-upto-def

by (intro sum-nonneg mult-nonneg-nonneg mangoldt-nonneg)

(auto simp: floor-le-iff)

moreover have x / real d ≤ 1 + real-of-int [x / real d] for d by linarith

hence r x ≤ sum-upto (λd. mangoldt d * 1) x unfolding sum-upto-altdef eq r-def using x

by (intro sum-mono mult-mono mangoldt-nonneg)

(auto simp: less-imp-le[OF frac-lt-1 algebra-simps])

ultimately show norm (r x) ≤ 1 * norm (ψ x) by (simp add: ψ-def)

qed auto

also have ψ ∈ O(λx. x) by (fact ψ-bigo)

finally have r: r ∈ O(λx. x).

define r' where r' = (λx::real. sum-upto ln x − x * ln x)

have r'-bigo: r' ∈ O(λx. x)

using stirling-weak-bigo unfolding r'-def.

have ln-fact: ln (fact n) = (Σ d=1..n. ln d) for n

by (induction n) (simp-all add: ln-mult)

hence r': sum-upto ln n = n * ln n + r'n for n :: real

unfolding r'-def sum-upto-altdef by (auto intro!: sum.cong)

have eventually (λn. sum-upto (λd. mangoldt d / d) n − ln n = r'n / n + r

n / n) at-top

using eventually-gt-at-top

proof eventually-elim

fix x :: real assume x: x > 0

have sum-upto ln x = sum-upto (λn. mangoldt n * real (nat [x / n])) x

unfolding sum-upto-ln-cons-sum-upto-mangoldt ..

also have . . = sum-upto (λd. mangoldt d * (x / d)) x − r x

unfolding sum-upto-def by (simp add: algebra-simps sum-subtractf r-def

sum-upto-def)

also have sum-upto (λd. mangoldt d * (x / d)) x = x * sum-upto (λd. mangoldt d / d) x

unfolding sum-upto-def by (subst sum-distrib-left) (simp add: field-simps)

finally have x * sum-upto (λd. mangoldt d / real d) x = r' x + r x + x * ln x

by (simp add: r' algebra-simps)

thus sum-upto (λd. mangoldt d / d) x − ln x = r' x / x + r x / x

using x by (simp add: field-simps)

qed
hence \((\lambda x. \text{sum-upto } (\lambda d. \text{mangoldt } d \div d) x - \ln x) \in \Theta(\lambda x. r' x / x + r x / x)\)
by \text{(rule bigthetaI-cong)}
also have \((\lambda x. r' x / x + r x / x) \in O(\lambda x. 1)\)
by \text{(intro sum-in-bigo) (insert } r' \text{-bigo, auto simp; landau-divide-simps)}
finally show ?thesis .
qed

Combining these two gives us Mertens’ first theorem.

\begin{isabelle}
\begin{isamath}
\textbf{theorem} \ mertens-bounded : \ (\lambda x. M' x - \ln x) \in O(\lambda x. 1)\\
\textbf{proof} -\\
\textbf{define} f \ \textbf{where} \ f = \text{sum-upto } (\lambda d. \text{mangoldt } d \div d)\\
\textbf{have} (\lambda x. (f x - \ln x) - (f x - M' x)) \in O(\lambda x. 1)\\
\textbf{using} \ \text{sum-upto-mangoldt-over-id-asymptotics}\\
\text{sum-upto-mangoldt-over-id-minus-phi-bounded}\\
\textbf{unfolding} f-def \ \textbf{by} \ (\text{rule sum-in-bigo})\\
\textbf{thus} \ ?thesis \ \textbf{by} \ simp\\
\end{isamath}
\end{isabelle}

qed

\begin{isabelle}
\begin{isamath}
\textbf{lemma} \ primes-M-bigo : M' \in O(\lambda x. \ln x)\\
\textbf{proof} -\\
\textbf{have} (\lambda x. M' x - \ln x) \in O(\lambda x. 1)\\
\textbf{by} \ (\text{rule mertens-bounded})\\
\textbf{also have} (\lambda x. x) \in O(\lambda x. \ln x)\\
\textbf{by} \ real-asymptotic\\
\textbf{finally have} (\lambda x. M' x - \ln x + \ln x) \in O(\lambda x. \ln x)\\
\textbf{by} \ (\text{rule sum-in-bigo}) \ \text{auto}\\
\textbf{thus} \ ?thesis \ \textbf{by} \ simp\\
\end{isamath}
\end{isabelle}

qed

end

4 The Prime Number Theorem

\textbf{theory} \ Prime-Number-Theorem
\textbf{imports} \ Newman-Ingham-Tauberian \ Prime-Counting-Functions
begin

4.1 Constructing Newman’s function

Starting from Mertens’ first theorem, i.e. \(M'(x) = \ln x + O(1)\), we now want
to derive that \(M'(x) = \ln x + c + o(1)\). This result is considerably stronger
and it implies the Prime Number Theorem quite directly.
In order to do this, we define the Dirichlet series

\[ f(s) = \sum_{n=1}^{\infty} \frac{\mathcal{M}(n)}{n^s}. \]

We will prove that this series extends meromorphically to \( \Re(s) \geq 1 \) and apply Ingham’s theorem to it (after we subtracted its pole at \( s = 1 \)).

**definition** \( \text{fds-newman where} \)

\( \text{fds-newman} = \text{fds} (\lambda n. \text{complex-of-real} (\Re n)) \)

**lemma** \( \text{fds-nth-newman:} \)

\( \text{fds-nth \text{fds-newman} \, n = of-real} (\Re n) \)

by \( (\text{simp add: \text{fds-newman-def \text{fds-nth-fds}}}) \)

**lemma** \( \text{norm-fds-nth-newman:} \)

\( \text{norm (fds-nth \text{fds-newman} \, n) = \Re n} \)

**unfolding** \( \text{fds-nth-newman norm-of-real} \)

by \( (\text{intro \text{abs-of-nonneg sum-nonneg divide-nonneg-pos}}) \)

(auto dest: prime-ge-1-nat)

The Dirichlet series \( f(s) + \zeta'(s) \) has the coefficients \( \Re(n) - \ln n \), so by Mertens’ first theorem, \( f(s) + \zeta'(s) \) has bounded coefficients.

**lemma** \( \text{bounded-coeffs-newman-minus-deriv-zeta:} \)

defines \( f \equiv \text{fds-newman} + \text{fds-deriv \text{fds-zeta}} \)

shows \( \text{Bseq} (\lambda n. \text{fds-nth \, f \, n}) \)

proof –

have \( (\lambda n. \Re (\text{real} n) - \ln (\text{real} n)) \in O(\lambda-. 1) \)

using \( \text{mertens-bounded} \) by \( (\text{rule landau-o.big.compose}) \)

real-asym

from \( \text{natfun-bigo-1E[OF this, of 1]} \)

obtain \( c \) where \( c: c \geq 1 \bigwedge n. |\Re (\text{real} n) - \ln (\text{real} n)| \leq c \) by auto

show \( \text{thesis} \)

proof \( (\text{intro \text{BseqI[of c] allII}}) \)

fix \( n : \text{nat} \)

show \( \text{norm (fds-nth f \, n) \leq c} \)

proof \( (\text{cases \, n = 0}) \)

case False

hence \( \text{fds-nth \, f \, n = of-real (\Re n - \ln n)} \)

by \( (\text{simp add: \text{f-def \text{fds-nth-newman \text{fds-nth-deriv \text{fds-nth-zeta}} scaleR-conv-of-real}}}) \)

also from \( (n \neq 0) \) have \( \text{norm \ldots \leq c} \)

using \( c(2)[of \, n] \) by \( (\text{simp add: \text{in-Reals-norm}}) \)

finally show \( \text{thesis} \).

qed (insert c, auto)

qed (insert c, auto)

qed

A Dirichlet series with bounded coefficients converges for all \( s \) with \( \Re(s) > 1 \) and so does \( \zeta'(s) \), so we can conclude that \( f(s) \) does as well.

**lemma** \( \text{abs-conv-abscissa-newman: abs-cone-abscissa \, f\text{-newman} \leq 1} \)

87
and \(\text{conv-abscissa-newman} : \text{conv-abscissa fds-newman} \leq 1\)

**proof**

- **define** \(f\) **where** \(f = \text{fds-newman} + \text{fds-deriv fds-zeta}\)
- **have** \(\text{abs-conv-abscissa } f \leq 1\)
  - **using** \(\text{bounded-coeffs-newman-minus-deriv-zeta}\) **unfolding** \(f\)-**def**
  - **by** (rule \(\text{bounded-coeffs-imp-abs-conv-abscissa-le-1}\))
- **hence** \(\text{abs-conv-abscissa } (f - \text{fds-deriv } \text{fds-zeta}) \leq 1\)
- **also have** \(f - \text{fds-deriv } \text{fds-zeta} = \text{fds-newman}\) **by** (simp add: \(f\)-**def**)
- **finally show** \(\text{abs-conv-abscissa } \text{fds-newman} \leq 1\)

**qed**

We now change the order of summation to obtain an alternative form of \(f(s)\) in terms of a sum of Hurwitz \(\zeta\) functions.

**lemma** \(\text{eval-fds-newman-conv-infsetsum}\):

- **assumes** \(s : \Re s > 1\)
- **shows** \(\text{eval-fds } \text{fds-newman } s = (\sum \{ a p | \text{prime } p \} . (\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \ s)\)
  - \((\lambda p . \ln (\text{real } p) / \text{real } p * \text{hurwitz-zeta } p \ s)\) **abs-summable-on** \(\{ p . \text{prime } p \}\)

**proof**

- **from** \(s\) **have conv: \(\text{fds-conv-deriv}\) \(\text{fds-newman}\) \(s\)
  - **by** (intro \(\text{fds-conv-deriv conv-le-less-trans}[OF \text{abs-conv-abscissa-newman}]\)) auto
- **define** \(f\) **where** \(f = (\lambda n p . \ln (\text{real } p) / \text{real } p / \text{of-nat } n ^{\text{powr } s})\)
- **have eq: \((\sum \{ a n \in \{p..\} . f n p \}) = \ln (\text{real } p) / \text{real } p * \text{hurwitz-zeta } p \ s\) if prime p for p\)
  - **by** (simp add: \(f\)-**def**)
  - **also have** \(\ldots = (\ln (\text{real } p) / \text{of-nat } p) * (\sum \{ a x \in \{p..\} . (\ln (\text{real } p) / \text{of-nat } p) * (1 / \text{of-nat } x ^{\text{powr } s})\})\)
    - **by** (simp add: \(f\)-**def**)
    - **also have** \((\sum \{ a x \in \{p..\} . 1 / \text{of-nat } x ^{\text{powr } s}) = \text{hurwitz-zeta } p \ s\)\)
      - **using** \(\text{abs-summable-hurwitz-zeta[of } s 0 \text{] that } s\)
      - **by** (intro infsetsum-cmult-right) (auto dest: \(\text{prime-gt-0-nat}\))
      - **also have** \((\sum \{ a x \in \{p..\} . 1 / \text{of-nat } x ^{\text{powr } s}) = \text{hurwitz-zeta } p \ s\)\)
        - **using** \(s\) **that** **by** (subst \(\text{hurwitz-zeta-nat-conv-infsetsum}(2)\))
          - (auto dest: \(\text{prime-gt-0-nat simp: field-simps powr-minus}\))
  - **finally show** \(?thesis\) .
- **qed**

**have norm-f: norm \((f n p) = \ln p / p / n ^{\text{powr } \Re s}\) if prime p for n p :: nat**
  - **by** (auto simp: \(f\)-**def norm-divide norm-mult norm-powr-real-powr\))
- **from conv have** \((\lambda n. \text{norm } (\text{fds-nth } \text{fds-newman } n / n ^{\text{powr } s}))\) **abs-summable-on** \(\text{UNIV}\)
  - **by** (intro \(\text{abs-summable-on-normal1}\)) (simp add: \(\text{fds-conv-deriv-altdef'}\))
  - **also have** \((\lambda n. \text{norm } (\text{fds-nth } \text{fds-newman } n / n ^{\text{powr } s})) = (\lambda n. \sum \{ p | \text{prime } p \land p \leq n. \text{norm } (f n p))\)
    - **by** (auto simp: \(\text{norm-divide norm-fds-nth-newman sum-divide-distrib primes-M-def}\)

88
prime-sum-upto-def norm-mul norm-f norm-powr-real-powr intro:

finally have summable1: (λ(n,p). f n p) abs-summable-on (SIGMA n:UNIV. 
{p. prime p ∧ p ≤ n})
using conv by (subst abs-summable-on-Sigma-iff) auto
also have ?this (-→ (λ(n,p). f n p) abs-summable-on
(λ(n,p). (p, n)) ) (SIGMA n:UNIV. 
{p. prime p ∧ p ≤ n})
by (subst abs-summable-on-reindex-iff [symmetric]) (auto simp: case-prod-unfold
inj-on-def)
also have (λ(n,p). (p, n)) (SIGMA n:UNIV. 
{p. prime p ∧ p ≤ n}) =
(SIGMA p:{p. prime p}. 
{p..}) by auto
finally have summable2: (λ(n,p). f n p) abs-summable-on . . .
from abs-summable-on-Sigma-project1 [OF this]
have (λp. ∑a(n, p) ∈{p..}. f n p) abs-summable-on 
{p. prime p} by auto
also have ?this (-→ (λp. ln (real p) / real p * hurwitz-zeta p s) abs-summable-on
{p. prime p})
by (intro abs-summable-on-cong eq) auto
finally show . . .

have eval-fds fds-newman s =
(∑a n, ∑a p | prime p ∧ p ≤ n. ln (real p) / real p / of-nat n powr s)
using conv by (simp add: eval-fds-altdef fds-nth-newman sum-divide-distrib
primes-M-def prime-sum-upto-def)
also have . . . = (∑a n, ∑a p | prime p ∧ p ≤ n. f n p)
unfolding f-def by (subst infsetsum-finite) auto
also have . . . = (∑a(n, p) ∈ (SIGMA n:UNIV. 
{p. prime p ∧ p ≤ n}). f n p)
using summable1 by (subst infsetsum-Sigma) auto
also have . . . = (∑a(p, n) ∈ (λ(n,p), (p, n)) (SIGMA n:UNIV. 
{p. prime p ∧ p ≤ n}), f n p)
by (auto simp: case-prod-unfold inj-on-def)
also have (λ(n,p). (p, n)) (SIGMA n:UNIV. 
{p. prime p ∧ p ≤ n}) =
(SIGMA p:{p. prime p}. 
{p..}) by auto
also have (∑a(p, n) ∈ . . . f n p) = (∑a p | prime p. ∑a n ∈{p..}, f n p)
using summable2 by (subst infsetsum-Sigma) auto
also have (∑a p | prime p. ∑a n ∈{p..}, f n p) =
(∑a p | prime p. ln (real p) / real p * hurwitz-zeta p s)
by (intro infsetsum-cong eq) auto
finally show eval-fds fds-newman s =
(∑a p | prime p. ln (real p) / real p * hurwitz-zeta p s).

qed

We now define a meromorphic continuation of \( f(s) \) on \( \Re(s) > \frac{1}{2} \).

To construct \( f(s) \), we express it as

\[
    f(s) = \frac{1}{z - 1} \left( \bar{f}(s) - \frac{\zeta'(s)}{\zeta(s)} \right),
\]

where \( \bar{f}(s) \) (which we shall call \( \text{pre-newman} \)) is a function that is analytic
on \( \Re(s) > \frac{1}{2} \), which can be shown fairly easily using the Weierstra M test.
\( \zeta'(s)/\zeta(s) \) is meromorphic except for a single pole at \( s = 1 \) and one \( k \)-th order pole for any \( k \)-th order zero of \( \zeta \), but for the Prime Number Theorem, we are only concerned with the area \( \Re(s) \geq 1 \), where \( \zeta \) does not have any zeros.

Taken together, this means that \( f(s) \) is analytic for \( \Re(s) \geq 1 \) except for a double pole at \( s = 1 \), which we will take care of later.

**context**

\[\text{fixes } A :: \text{nat} \Rightarrow \text{complex} \Rightarrow \text{complex} \quad \text{and } B :: \text{nat} \Rightarrow \text{complex} \Rightarrow \text{complex}\]

\[\text{defines } A \equiv (\lambda p \ s \ . \ (s - 1) \ast \text{pre-zeta} (\text{real } p) \ s \ - \ \text{of-nat } p \ / \ (\text{of-nat } p \ \text{powr } s * (\text{of-nat } p \ \text{powr } s - 1)))\]

\[\text{defines } B \equiv (\lambda p \ s \ . \ \text{of-real} (\ln (\text{real } p)) / \text{of-nat } p \ast A p s)\]

begin

**definition** pre-newman :: complex \Rightarrow complex where

\[\text{pre-newman } s = (\sum p . \text{if prime } p \text{ then } B p s \text{ else } 0)\]

**definition** newman where

\[\text{newman } s = 1 / (s - 1) \ast (\text{pre-newman } s - \text{deriv} \ \zeta \ s / \zeta \ s)\]

The sum used in the definition of pre-newman converges uniformly on any disc within the half-space with \( \Re(s) > \frac{1}{2} \) by the Weierstra M test.

**lemma** uniform-limit-pre-newman:

**assumes** \( r : r \geq 0 \ \Re s - r > 1 / 2 \)

**shows** uniform-limit (cball s r)

\[\quad (\lambda n \ s . \ \sum \text{if } \text{prime } p \text{ then } B p s \text{ else } 0)\ \text{pre-newman at-top}\]

**proof**

- **from** \( r \) **have** \( \Re: \Re z > 1 / 2 \) **if** \( \text{dist } s \ z \leq r \) **for** \( z \)
  - **using** abs-Re-le-cmod[of s - z] \( r \) **that**
    - **by** (auto simp: dist-norm abs-if split: if-splits)

**define** \( x \) **where** \( x = \Re s - r \) — The lower bound for the real part in the disc

- **from** \( r \) **Re have** \( x > 1 / 2 \) **by** (auto simp: x-def)

  — The following sequence \( M \) bounds the summand, and it is obviously \( O(n^{-1-\epsilon}) \) and therefore summable

**define** \( C \) **where** \( C = (\text{norm } s + r + 1) \ast (\text{norm } s + r) / x\)

**define** \( M \) **where** \( M = (\lambda p :: \text{nat} . \ln p \ast (C / \text{powr } (x + 1) + 1 / (\text{powr } x * (p \ \text{powr } x - 1))))\)

**show** \( \text{thesis unfolding } \text{pre-newman-def} \)

**proof** (intro weierstrass-m-test-ev[OF eventually-mono[OF eventually-gt-at-top[of 1]]] ballI)

**show** summable \( M \)

**proof** (rule summable-comparison-test-bigo)

**define** \( \epsilon \) **where** \( \epsilon = \text{min } (2 \ast x - 1) \ x / 2\)

**from** \( (x > 1 / 2) \) **have** \( \epsilon : \epsilon > 0 \ 1 + \epsilon < 2 \ast x 1 + \epsilon < x + 1 \)

**by** (auto simp: \( \epsilon \)-def min-def field-simps)

**show** \( M \in O(\lambda n. \ \text{powr } (- 1 - \epsilon)) \) **unfolding** \( \text{M-def distrib-left} \)

90
by (intro sum-in-bigO) (use ϵ in real-asympt)+
from ϵ show summable (λn. norm (p powr (− 1 − ϵ)))
  by (simp add: summable-real-powr-s)
qed

next

fix p :: nat and z assume p: p > 1 and z: z ∈ cball s r
from z r Re[of z] have x: Re z ≥ x x > 1 / 2 and Re z > 1 / 2
  using abs-Re-le-creds[of s − z] by (auto simp: x-def algebra-simps dist-norm)
have norm-z: norm z ≤ norm s + r
  using z norm-triangle-ineq2[of z s] r by (auto simp: dist-norm norm-minus-commute)
from p > 1 and x and r have M p ≥ 0
  by (auto simp: C-def M-def intro!: mult-nonneg-nonneg add-nonneg-nonneg divide-nonneg-pos)

have bound: norm ((z − 1) * pre-zeta p z) ≤
  norm (z − 1) * (norm z / (Re z * p powr Re z))
  using pre-zeta-bound "[of z p] p : Re z > 1 / 2;"
unfolding norm-mult by (intro mult-mono pre-zeta-bound) auto

have norm (B p z) = ln p / p * norm (A p z)
unfolding B-def using (p > 1) by (simp add: B-def norm-mult norm-divide)
also have ... ≤ ln p / p * (norm (z − 1) * norm z / Re z / p powr Re z +
  p / (p powr Re z * (p powr Re z − 1)))
unfolding A-def using (p > 1; and (Re z > 1 / 2; and bound)
  by (intro mult-left-mono order.trans[OF norm-triangle-ineq4 add-mono]
    mult-left-mono)
  (auto simp: norm-divide norm-mult norm-powr-real-powr
    intro!: divide-left-mono order.trans[OF - norm-triangle-ineq2])
  also have ... = ln p * (norm (z − 1) * norm z / Re z / p powr (Re z + 1)) +
  1 / (p powr Re z * (p powr Re z − 1)))
  using (p > 1) by (simp add: field_simps powr-add powr-minus)
  also have norm (z − 1) * norm z / Re z / p powr (Re z + 1) ≤ C / p powr
  (x + 1)
  unfolding C-def using r : Re z > 1 / 2; norm-z p x
  by (intro mult-mono frac-le powr-mono order.trans[OF norm-triangle-ineq4])
auto
  also have 1 / (p powr Re z * (p powr Re z − 1)) ≤
  1 / (p powr x * (p powr x − 1)) using (p > 1) x
  by (intro divide-left-mono mult-mono powr-mono diff-right-mono mult-pos-pos)
  (auto simp: ge-one-powr-ge-zero)
  finally have norm (B p z) ≤ M p
  using (p > 1) by (simp add: mult-left-mono M-def)
  with (M p ≥ 0) show norm (if prime p then B p z else 0) ≤ M p by simp
qed

lemma sums-pre-newman: Re s > 1 / 2 ⇒ (λp. if prime p then B p s else 0)
  sums pre-newman s
using \textit{tendsto-uniform-limit}\{OF uniform-limit-pre-newman[of 0 s]\} by (auto simp: sums-def)

lemma analytic-pre-newman \{THEN analytic-on-subset, analytic-intros\}:
  \(\text{pre-newman analytic-on \{s. Re s > 1 / 2\}}\)

proof
  have \textit{holo} \((\lambda s::\text{complex}. \text{ if prime } p \text{ then } B p s \text{ else } 0) \text{ holomorphic-on } X\)
    if \(X \subseteq \{s. Re s > 1 / 2\}\) \textit{for } \(X\) \textit{and } \(p :: \text{nat}\) \textit{using } that
    by (cases prime p)
      (auto intro!: holomorphic-intros simp: B-def A-def dest \(\prime\): prime-gt-1-nat)
  have \textit{holo'} \(\text{pre-newman holomorphic-on ball s r } \text{if } r > 0 \text{ Re s - r > 1 / 2}\)
    for \(s r\)
  proof
    have Re: \(\text{Re } z > 1 / 2\) \textit{if } \(\text{dist s z } \leq r\) \textit{for } \(z\)
      using \textit{abs-Re-le-cmod[of s - z]} \(r\) \textit{that } by (auto simp: dist-norm abs-if split: if-splits)
    show \?thesis
      by (rule holomorphic-uniform-limit\{OF - uniform-limit-pre-newman[of r s]\})
  qed
  qed

lemma holomorphic-pre-newman \{holomorphic-intros\}:
  \(X \subseteq \{s. Re s > 1 / 2\} \implies \text{pre-newman holomorphic-on } X\)

using analytic-pre-newman \textit{by} (rule analytic-imp-holomorphic)

lemma eval-fds-newman:
  assumes s: \(\text{Re } s > 1\)
  shows \(\text{eval-fds fds-newman } s = \text{newman } s\)

proof
  have eq: \((\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \text{ s } =\)
    \(1 / (s - 1) * (\text{ln (real } p) / (p \text{ powr } s - 1) + B p s)\)
  if p: \text{prime } p \text{ for } p
  proof
    have \((\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \text{ s } =\)
      \(\ln (\text{real } p) / \text{real } p * (p \text{ powr } (1 - s) / (s - 1) + \text{pre-zeta } p \text{ s})\)
    using s \textit{by} (auto simp add: hurwitz-zeta-def)
    also have \(\ldots = 1 / (s - 1) * (\text{ln (real } p) / (p \text{ powr } s - 1) + B p s)\)
      using p s \textit{by} (simp add: divide-simps powr-diff B-def)
    \(\text{auto simp: A-def field-simps dest: prime-gt-1-nat}\)
    finally show \?thesis.
  qed
have (λp. (ln (real p) / real p) * hurwitz-zeta p s) abs-summable-on {p. prime p}
  using s by (intro eval-fds-newman-conv-infsetsum)

hence (λp. 1 / (s - 1) * (ln (real p) / (p powr s - 1) + B p s))
  abs-summable-on {p. prime p}
by (subst (asm) abs-summable-on-cong (OF eq refl)) auto

hence summable:
(lam ln (real p) / (p powr s - 1) + B p s) abs-summable-on {p. prime p}
using s by (subst (asm) abs-summable-on-cmult-right-iff) auto

from s have [simp]: s ≠ 1 by auto
have eval-fds fds-newman s =
  (∑a p | prime p. (ln (real p) / real p) * hurwitz-zeta p s)
  using s by (rule eval-fds-newman-conv-infsetsum)
also have ... = (∑a p | prime p. 1 / (s - 1) * (ln (real p) / (p powr s - 1) + B p s))
  by (intro infsetsum-cong eq) auto
also have ... = 1 / (s - 1) * (∑a p | prime p. ln (real p) / (p powr s - 1) + B p s)
  (is _ = - * ?S) by (rule infsetsum-cmult-right (OF summable))
also have ?S = (∑p. if prime p then
    ln (real p) / (p powr s - 1) + B p s else 0)
  by (subst infsetsum-nat (OF summable)) auto
also have ... = (∑p. if prime p then ln (real p) / (p powr s - 1) else 0) +
  (if prime p then B p s else 0))
  by (intro suminf-cong) auto
also have ... = pre-newman s - deriv zeta s / zeta s
  using sums-pre-newman [of s] sums-logderiv-zeta [of s] s
  by (subst suminf-add [symmetric]) (auto simp: sums-iff)
finally show ?thesis by (simp add: Newman-def)
qed
end

Next, we shall attempt to get rid of the pole by subtracting suitable multiples of \(\zeta(s)\) and \(\zeta'(s)\). To this end, we shall first prove the following alternative definition of \(\zeta'(s)\):

**lemma deriv-zeta-eq:**

assumes 0 < Re s s ≠ 1

shows deriv zeta s = deriv (λz. pre-zeta 1 z * (z - 1)) s / (s - 1) -
  (pre-zeta 1 s * (s - 1) + 1) / (s - 1)^2

(is _ = ?rhs)

**proof** (rule DERIV-imp-deriv)

have [derivative-intros]: (pre-zeta 1 has-field-derivative deriv (pre-zeta 1) s) (at s)
  by (intro holomorphic-deriv [of - UNIV] holomorphic-intros) auto

have *: deriv (λz. pre-zeta 1 z * (z - 1)) s = deriv (pre-zeta 1) s * (s - 1) +
  pre-zeta 1 s

93
proof
(auto intro!: holomorphic-on-imp-differentiable-at[of - UNIV] holomorphic-intros)

hence 
\((\lambda s. \text{pre-zeta } 1 s + 1 / (s - 1)) \text{ has-field-derivative}\)
\(\text{deriv (pre-zeta } 1 s - 1 / ((s - 1) \times (s - 1)))\) (at s)
using assms by (auto intro!: derivative-eq-intros)
also have \(\text{deriv (pre-zeta } 1 s - 1 / ((s - 1) \times (s - 1)) = ?rhs}\)
using * assms by (simp add: divide-simps) (auto simp: field-simps power2-eq-square)
also have \((\lambda s. \text{pre-zeta } 1 s + 1 / (s - 1)) \text{ has-field-derivative } ?rhs\) (at s) \(\iff\)
(zeta has-field-derivative ?rhs) (at s)
using assms
by (intro has-field-derivative-cong-ev eventually-mono[OF t1-space-nhds[of - 1]])
(auto simp: zeta-def hurwitz-zeta-def)
finally show \ldots .

qed

From this, it follows that \((s - 1)ζ'(s) - ζ'(s)/ζ(s)\) is analytic for \(\Re(s) \geq 1\):

lemma analytic-zeta-derivdiff:

obtains a where
\((λz. \text{if } z = 1 \text{ then } a \text{ else } (z - 1) \times \text{ deriv zeta } z / \text{ zeta } z)\)
analytic-on \([s. \Re s \geq 1]\)

proof

have neq: \(\text{pre-zeta } 1 z \times (z - 1) + 1 \neq 0\) if \(\Re z \geq 1\) for z
using zeta-Re-ge-1-nonzero[of z] that
by (cases z = 1) (auto simp: zeta-def hurwitz-zeta-def divide-simps)
let \(?g = λz. (1 - inverse (\text{pre-zeta } 1 z \times (z - 1) + 1)) \times ((z - 1) \times \text{ deriv (λx. pre-zeta } 1 u \times (u - 1))) z = (\text{pre-zeta } 1 z \times (z - 1) + 1))\)

show \((λz. \text{if } z = 1 \text{ then } ?g 1 \text{ else } (z - 1) \times \text{ deriv zeta } z - \text{ deriv zeta } z / \text{ zeta } z)\)
analytic-on \([s. \Re s \geq 1]\) (is \?f analytic-on -)

proof (rule pole-theorem-analytic-0)

show \(?g \text{ analytic-on } [s. 1 \leq \Re s]\) using neq
by (auto intro!: analytic-intros)

next

show \(∃d>0. ∀ w ∈ \text{ball } z d - \{1\}. ?g w = (w - 1) \times \?f w\)
if \(z: z \in \{s. 1 \leq \Re s\}\) for z

proof -

have *: \(\text{isCont } (\lambda z. \text{pre-zeta } 1 z \times (z - 1) + 1) z\)
by (auto intro!: continuous-intros)

obtain e where e > 0 and e: \(\exists y. \text{dist } z y < e \implies \text{pre-zeta } (\text{Suc } 0) y * (y - 1) + 1 \neq 0\)
using continuous-at-avoid [OF * neq[of z]] z by auto

show ?thesis

proof (intro exI ballI conjI)

fix \(w\)
assume w: \(w \in \text{ball } z (\text{min } e \times 1) - \{1\}\)
then have \(\Re w > 0\)
using complex-Re-le-cmod [of \(-w\)] z by (simp add: dist-norm)

with \(w\) show \(?g w = (w - 1) \times (\text{if } w = 1 \text{ then } \text{deriv } ?g 1 \text{ else})\)

94
Finally, \( f(s) + \zeta'(s) + c\zeta(s) \) is analytic.

**Lemma analytic-newman-variant:**

- **Obtains** \( c \) \( a \) where
  - \( (\lambda z. \text{if } z = 1 \text{ then } a \text{ else } \text{newman } z + \text{deriv } zeta \ z + c * \text{zeta } z) \) analytic-on\( \{ s. \text{Re } s \geq 1 \} \)

**Proof**

- **Obtain** \( c \) \( d \) where
  - \( c: (\lambda z. \text{if } z = 1 \text{ then } c \text{ else } (z - 1) * \text{deriv } zeta \ z - \text{deriv } zeta \ z / \text{zeta } z) \) analytic-on\( \{ s. \text{Re } s \geq 1 \} \)

  **Using** analytic-zeta-derivdiff by blast

**Let** \( ?g = \lambda z. \text{pre-newman } z + (\text{if } z = 1 \text{ then } c \text{ else } (z - 1) * \text{deriv } zeta \ z - \text{deriv } zeta \ z / \text{zeta } z) - (c + \text{pre-newman } 1) * (\text{pre-zeta } 1 \ z * (z - 1) + 1) \)

**Have** \( (\lambda z. \text{if } z = 1 \text{ then } \text{deriv } ?g \ 1 \text{ else } \text{newman } z + \text{deriv } zeta \ z + (-(c + \text{pre-newman } 1)) * \text{zeta } z) \) analytic-on\( \{ s. \text{Re } s \geq 1 \} \) (is \( ?f \) analytic-on -)

**Proof** (rule pole-theorem-analytic-0)

**Show** \( ?g \) analytic-on\( \{ s. 1 \leq \text{Re } s \} \)

**By** (intro \( c \) analytic-intros) auto

**Next**

**Show** \( \exists d > 0. \forall w \in \text{ball } z \ d - \{1\}. \ ?g \ w = (w - 1) * ?f \ w \)

**If** \( z \in \{ s. 1 \leq \text{Re } s \} \) for \( z \) using that

**By** (intro \( \text{ex} \{ of \ - \1 \}, \text{simp-all add: } \text{newman-def divide-simps } \text{zeta-def hurwitz-zeta-def} \))

**Auto simp: field-simps**?

**Qed**

**With** that show \( ?\)thesis by blast

**Qed**

### 4.2 The asymptotic expansion of \( \mathcal{M} \)

Our next goal is to show the key result that \( \mathcal{M}(x) = \ln n + c + o(1) \).

As a first step, we invoke Ingham’s Tauberian theorem on the function we have just defined and obtain that the sum

\[
\sum_{n=1}^{\infty} \frac{\mathcal{M}(n) - \ln n + c}{n}
\]

exists.
lemma mertens-summable:
  obtains c :: real where summable (λn. (ℜ n - Ln n + c) / n)
proof -

from analytic-newman-variant obtain c a where
  analytic: (λz. if z = 1 then a else newman z + deriv zeta z + c * zeta z)
  define f where f = (λz. if z = 1 then a else newman z + deriv zeta z + c * zeta z)

have analytic: f analytic-on {s. Re s ≥ 1} using analytic by (simp add: f-def)
define F where F = fds-newman + fds-deriv fds-zeta + fds-const c * fds-zeta

note le = conv-abcissa-add-leI conv-abcissa-deriv-le conv-abcissa-newman conv-abcissa-mult-const-left
note intros = le le[THEN le-less-trans] le[THEN order.trans] fds-converges
have eval-F: eval-fds F s = f s if s: Re s ≥ 1 for s
proof -
  have eval-fds F s = eval-fds (fds-newman + fds-deriv fds-zeta) s +
                 eval-fds (fds-const c * fds-zeta) s
    unfolding F-def using s by (subst eval-fds-add) (auto intro!: intros)
  also have ... = f s using s unfolding f-def
    by (subst eval-fds-add)
      (auto intro!: intros simp: eval-fds-newman eval-fds-deriv-zeta eval-fds-mult eval-fds-zeta)
  finally show ?thesis .
qed

have conv: fds-converges F s if Re s ≥ 1 for s
proof (rule Newman-Ingham)
  have (λn. ℜ (real n) - Ln (real n)) ∈ O(λ-. 1)
    using mertens-bounded by (rule landau-o.big.compose) real-asymp
from natfun-bigo-1E[OF this, of 1]
  obtain c' where c': c' ≥ 1 ∧ n. ℜ (real n) - Ln (real n) ≤ c' by auto
have Bseq (fds-nth F)
proof (intro BseqI allI)
  fix n :: nat
  show norm (fds-nth F n) ≤ (c' + norm c) unfolding F-def using c'
    by (auto simp: fds-nth-zeta fds-nth-deriv fds-nth-newman scaleR-conv-of-real in-Reals-norm
        intro!: order.trans[OF norm-triangle-ineq] add-mono)
    qed (insert c', auto intro: add-pos-nonneg)
  thus fds-nth F ∈ O(λ-. 1) by (simp add: natfun-bigo-iff-Bseq)
next
  show f analytic-on {s. Re s ≥ 1} by fact
next
  show eval-fds F s = f s if Re s > 1 for s using that by (rule eval-F)
  qed (insert that, auto simp: F-def intro!: intros)
from conv[of 1] have summable (λn. fds-nth F n) / (of-nat n)
  unfolding fds-converges-def by auto
also have ?this ←→ summable (λn. (ℜ n - Ln n + c) / n)

96
Next, we prove a lemma given by Newman stating that if the sum \( \sum a_n/n \) exists and \( a_n + \ln n \) is nondecreasing, then \( a_n \) must tend to 0. Unfortunately, the proof is rather tedious, but so is the paper version by Newman.

**Lemma** sum-goestozero-lemma:

**Fixes** \( d : \text{real} \)

**Assumes** \( d : \lfloor \sum i = M..n. a i / i \rfloor < d \) and \( le: \forall n. a n + \ln n \leq a (\text{Suc} \ n) + \ln (\text{Suc} \ n) \)

**And** \( 0 < M \leq N \)

**Shows** \( a M \leq d * N / (\text{real} \ N - \text{real} \ M) + (\text{real} \ N - \text{real} \ M) / M \land -a N \leq d * N / (\text{real} \ N - \text{real} \ M) + (\text{real} \ N - \text{real} \ M) / M \)

**Proof**

1. **Have** \( 0 \leq d \)
   **Using** \( \text{assms by linarith} + \)

2. **Then have** \( 0 \leq d * N / (N - M + 1) \) by \( \text{simp} \)

3. **Then have** \( \text{le-dN:} [0 \leq x \implies x \leq d * N / (N - M + 1)] \implies x \leq d * N / (N - M + 1) \) for \( x : \text{real} \)
   **By** \( \text{linarith} \)

4. **Have** \( \text{le-a-ln:} a m + \ln m \leq a n + \ln n \) if \( n \geq m \) for \( n \ m \)
   **By** \( \text{rule transitive-stepwise-le} \) (use \( \text{le that in auto} \))

5. **Have** \( \forall b \land y \leq b \) if \( a \leq b \ x \leq a \ y \leq a \) for \( a \ b \ x \ y : \text{real} \)
   **Using** \( \text{that by linarith} \)

6. **Show** \( \text{thesis} \)

**Proof** (rule \( * \))

7. **Show** \( d * N / (N - M) + \ln (N / M) \leq d * N / (\text{real} \ N - \text{real} \ M) + (\text{real} \ N - \text{real} \ M) / M \)
   **Using** \( \theta < M, \ M < N, \ \text{ln-le-minus-one} \ [\text{of} \ N / M] \)
   **By** \( \text{simp add: of-nat-diff} (\text{simp add: divide-simps}) \)

**Next**

8. **Have** \( a M - \ln (N / M) \leq (d * N) / (N - M + 1) \)

**Proof** (rule \text{le-dN})

9. **Assume** \( 0 \leq a M - \ln (N / M) \)

10. **Have** \( \text{Suc} N - M \) * \( (a M - \ln (N / M)) / N = (\sum i = M..N. (a M - \ln (N / M)) / N) \)
    **By** \( \text{simp} \)

11. **Also have** … \( \leq (\sum i = M..N. a i / i) \)

**Proof** (rule sum-mono)

12. **Fix** \( i \)

13. **Assume** \( i : i \in \{M..N\} \)
with \(0 < M\) have \(0 < i\) by auto
have \((a M - \ln (N / M)) / N \leq (a M - \ln (N / M)) / i\)
using \(0\) using \(i:\!0 < M\); by (simp add: frac-le-eq divide-simps mult-left-mono)
also have \(a M + \ln (\text{real} M) \leq a i + \ln (\text{real} N)\)
by (rule order_trans[OF le-a-ln[of M i]]) (use i assms in auto)
hence \((a M - \ln (N / M)) / i \leq a i / \text{real} i\)
using assms i by (intro divide-right-mono) (auto simp: ln-div field-simps)
finally show \((a M - \ln (N / M)) / \text{real} N \leq a i / \text{real} i\).
qed
finally have \(((\text{Suc} N) - M) * (a M - \ln (N / M)) / N \leq |\sum i = M..N. a i / i|\)
by simp
also have \(\ldots \leq d\) using \(d\) by simp
finally have \(((\text{Suc} N) - M) * (a M - \ln (N / M)) / N \leq d\).
then show \(\text{thesis}\)
using \((M < N)\) by (simp add: of-nat-diff field-simps)
qed
also have \(\ldots \leq d * N / (N - M)\)
using assms(1,4) by (simp add: field-simps)
finally show \(a M \leq d * N / (N - M) + \ln (N / M)\) by simp
next
have \(- a N - \ln (N / M) \leq (d * N) / (N - M + 1)\)
proof (rule le-dN)
assume \(0: 0 \leq - a N - \ln (N / M)\)
have \((\sum i = M..N. a i / i) \leq (\sum i = M..N. (a N + \ln (N / M)) / N)\)
proof (rule sum-mono)
fix \(i\)
assume \(i: i \in \{M..N\}\)
with \(i < M\) have \(0 < i\) by auto
have \(a i + \ln (\text{real} M) \leq a N + \ln (\text{real} N)\)
by (rule order_trans[OF le-a-ln[of i N]]) (use i assms in auto)
hence \(a i / i \leq (a N + \ln (N / M)) / i\)
using assms i by (intro divide-right-mono) (auto simp: field-simps)
\(\ln\text{-div}\)
also have \(\ldots \leq (a N + \ln (N / M)) / N\)
using \(i \geq 0\) \(0\) by (intro divide-left-mono-neg) auto
finally show \(a i / i \leq (a N + \ln (N / M)) / N\).
qed
also have \(\ldots = (\sum i = M..N. a i / i) \leq (\text{real} (\text{Suc} N) - \text{real} M) * (a N + \ln (N / M)) / N\)
by simp
finally have \(((\text{Suc} N) - M) * (a N + \ln (N / M)) / N \leq |\sum i = M..N. a i / i|\)
using \((M < N)\) by (simp add: of-nat-diff)
then have \(-((\text{real} (\text{Suc} N) - \text{real} M) * (a N + \ln (N / M)) / N \leq |\sum i = M..N. a i / i|\)
by linarith
also have \(\ldots \leq d\) using \(d\) by simp
finally have \(-((\text{real} (\text{Suc} N) - \text{real} M) * (a N + \ln (N / M)) / N \leq d\).
then show \(\text{thesis}\)

98
using \( M < N \) by (simp add: of-nat-diff field-simps)

qed

also have \( \ldots \leq d * N / \text{real} (N - M) \)
using \( 0 < M : (M < N) ; 0 \leq d \) by (simp add: field-simps)

finally show \( -a N \leq d * N / \text{real} (N - M) + \ln (N / M) \) by simp

qed

proposition sum-goestozero-theorem:

assumes \( \text{summ}: \text{summable} \ (\lambda i. a i / i) \)
and \( \text{le}: \bigwedge n. a n + \ln n \leq a (\text{Suc} n) + \ln (\text{Suc} n) \)
shows \( a \xrightarrow{i} 0 \)

proof (clarsimp simp: lim-sequentially)

fix \( r :: \text{real} \)

assume \( r > 0 \)

have \( *: \exists n0. \forall n \geq n0. |a n| < \varepsilon \) if \( \varepsilon: 0 < \varepsilon \varepsilon < 1 \) for \( \varepsilon \)

proof –

have \( 0 < (\varepsilon / 8)^2 \) using \( 0 < \varepsilon \) by simp

then obtain \( N0 \) where \( N0: \bigwedge m. m \geq N0 \Longrightarrow \text{norm} (\sum k=m..n. (\lambda i. a i / i) k) < (\varepsilon / 8)^2 \)

by (metis summable-partial-sum-bound summ)

obtain \( N1 \) where \( \text{real} N1 > 4 / \varepsilon \)

using \( \text{reals-Archimedean2[of} 4 / \varepsilon \] \varepsilon \) by auto

hence \( N1 \neq 0 \) and \( \text{real} N1 < \varepsilon / 4 \) using \( \varepsilon \)

by (auto simp: divide-simps mult-ac intro: Nat.gr0I)

have \( |a n| < \varepsilon \) if \( n: n \geq 2 * N0 + N1 + 7 \) for \( n \)

proof –

define \( k \) where \( k = \lfloor n * \varepsilon / 4 \rfloor \)

have \( n * \varepsilon / 4 > 1 \) and \( n * \varepsilon / 4 \leq n / 4 \) and \( n / 4 < n \)

using less-le-trans[OF N1, of n / N1 * \varepsilon / 4] \( \langle N1 \neq 0 \rangle\) \varepsilon \varepsilon by (auto simp: field-simps)

hence \( k: k > 0 \) \( 4 * k \leq n \) nat \( k < n \) \( (n * \varepsilon / 4) - 1 < k \) \( k \leq (n * \varepsilon / 4) \)

unfolding \( k\text{-def} \) by linarith+

have \( -a n < \varepsilon \)

proof –

have \( N0 \leq n - \text{nat} k \)

using \( n k \) by linarith

then have \( *: \lfloor \sum k = n - \text{nat} k \ldots a k / k \rfloor < (\varepsilon / 8)^2 \)

using \( N0 [of n - \text{nat} k n] \) by simp

have \( -a n \leq (\varepsilon / 8)^2 * n / (n \cdot \varepsilon / 4) + (n \cdot \varepsilon / 4) / (n - k) \)

using sum-goestozero-lemma [OF \( \text{le}, \text{THEN conjunct2} \] \( k \) by (simp add: of-nat-diff k-def)

also have \( \ldots < \varepsilon \)

proof –

have \( \varepsilon / 16 * n / k < 2 \)

using \( k \) by (auto simp: field-simps)

then have \( \varepsilon * (\varepsilon / 16 * n / k) < \varepsilon * 2 \)

99
using $\varepsilon$ mult-less-cancel-left-pos by blast
then have $(\varepsilon / 8)^2 \cdot n / k < \varepsilon / 2$
  by (simp add: field-simps power2-eq-square)
moreover have $k / (n - k) < \varepsilon / 2$
proof -
  have $(\varepsilon + 2) \cdot k < 4 \cdot k$ using $k \varepsilon$ by simp
  also have ... $\leq \varepsilon \cdot real\ n$ using $k$ by (auto simp: field-simps)
  finally show $\varepsilon$thesis using $k$ by (auto simp: field-simps)
qed
ultimately show $\varepsilon$thesis unfolding k-def by linarith
qed
finally show $\varepsilon$thesis.
qed
moreover have $a\ n < \varepsilon$
proof -
  have $N_0 \leq n$ using $n\ k$ by linarith
  then have $\forall k. \sum_{n} k = n + \text{nat}\ k. a\ k / k < (\varepsilon/8)^2$
    using $N_0 [of n\ n + \text{nat}\ k]$ by simp
  have $a\ n \leq (\varepsilon/8)^2 \cdot (n + \text{nat}\ k) / k + k / n$
    using sum-goestozero-lemma [OF $\le$, THEN conjunct1] $k$ by (simp add: of-nat-diff)
  also have ... $< \varepsilon$
  proof -
    have $4 \leq 28 \cdot real\-of-int\ k$ using $k$ by linarith
    then have $\varepsilon/16 \cdot n / k < 2$ using $k$ by (auto simp: field-simps)
    have $\varepsilon \cdot (\text{real}\ n + k) < 32 \cdot k$
      proof -
        have $\varepsilon \cdot n / 4 < k + 1$ by (simp mult.commue k-def)
        then have $\varepsilon \cdot n < 4 \cdot k + 4$ by (simp add: divide-simps)
        also have ... $\leq 8 \cdot k$ using $k$ by auto
        finally have $1: \varepsilon \cdot real\ n < 8 \cdot k$.
        have $2: \varepsilon \cdot k < k$ using $k$ by simp
          show $\varepsilon$thesis using $k$ add-strict-mono [OF 1 2] by (simp add: algebra-simps)
      qed
    then have $(\varepsilon / 8)^2 \cdot real\ (n + \text{nat}\ k) / k < \varepsilon / 2$
      using $\varepsilon\ k$ by (simp add: divide-simps mult-less-0-iff power2-eq-square)
    moreover have $k / n < \varepsilon / 2$
      using $k\ varepsilon$ by (auto simp: k-def field-simps)
    ultimately show $\varepsilon$thesis by linarith
     qed
    finally show $\varepsilon$thesis.
    qed
ultimately show $\varepsilon$thesis by force
    qed
then show $\varepsilon$thesis by blast
    qed
show $\exists n0. \forall n\geq n0. |a\ n| < r$
  using $* [of min\ r\ (1/5)] (\theta < r)$ by force
This leads us to the main intermediate result:

**lemma Mertens-convergent**: convergent $(\lambda n::\text{nat}. \mathcal{M} n - \ln n)$

**proof**

- obtain $c$ where $c$ : summable $(\lambda n. (\mathcal{M} n - \ln n + c) / n)$
  
  by (blast intro: mertens-summable)

then obtain $l$ where $l$: $(\lambda n. (\mathcal{M} n - \ln n + c) / n)$ sums $l$

by (auto simp: summable_def)

have $*: (\lambda n. \mathcal{M} n - \ln n + c) \longrightarrow 0$

by (rule sum-goestozero-theorem[OF $c$]) auto

hence $(\lambda n. \mathcal{M} n - \ln n) \longrightarrow -c$

by (simp add: tendsto-iff dist-norm)

thus $\therefore$ thesis by (rule convergentI)

**corollary $\mathcal{M}$-minus-ln-limit**:

obtains $c$ where $(\lambda x::\text{real}. \mathcal{M} x - \ln x) \longrightarrow c$ at-top

**proof**

from Mertens-convergent obtain $c$ where $(\lambda n. \mathcal{M} n - \ln n) \longrightarrow c$

by (auto simp: convergent_def)

hence 1: $(\lambda x::\text{real}. \mathcal{M} (\lfloor x \rfloor) - \ln (\lfloor x \rfloor)) \longrightarrow c$ at-top

by (rule filterlim-compose real-asym)

have 2: $(\lambda x::\text{real}. \ln (\lfloor x \rfloor) - \ln x) \longrightarrow 0$ at-top

by real-asym

have 3: $(\lambda x. \mathcal{M} x - \ln x) \longrightarrow c$ at-top

using tendsto-add[OF 1 2] by simp

with that show $\therefore$ thesis by blast

**qed**

### 4.3 The asymptotics of the prime-counting functions

We will now use the above result to prove the asymptotics of the prime-counting functions $\vartheta(x) \sim x$, $\psi(x) \sim x$, and $\pi(x) \sim x/\ln x$. The last of these is typically called the Prime Number Theorem, but since these functions can be expressed in terms of one another quite easily, knowing the asymptotics of any of them immediately gives the asymptotics of the other ones.

In this sense, all of the above are equivalent formulations of the Prime Number Theorem. The one we shall tackle first, due to its strong connection to the $\mathcal{M}$ function, is $\vartheta(x) \sim x$.

We know that $\mathcal{M}(x)$ has the asymptotic expansion $\mathcal{M}(x) = \ln x + c + o(1)$. We also know that

$$\vartheta(x) = x\mathcal{M}(x) - \int_2^x \mathcal{M}(t) \, dt.$$
Substituting in the above asymptotic equation, we obtain:
\[
\vartheta(x) = x \ln x + cx + o(x) - \int_2^x \ln t + c + o(1) \, dt
\]
\[
= x \ln x + cx + o(x) - (x \ln x - x + cx + o(x))
\]
\[
= x + o(x)
\]
In conclusion, \( \vartheta(x) \sim x \).

**Theorem** \( \vartheta \)-asymptotics: \( \vartheta \sim \text{[at-top]} (\lambda x. x) \)

**Proof**

From \( M \)-minus-\( ln \)-limit obtain \( c \) where \( c \) is defined as \((\lambda x. M x - \ln x) \longrightarrow c\) at-top by auto.

Define \( r \) where \( r = (\lambda x. M x - \ln x - c) \)

Have \( M \)-expand: \( M = (\lambda x. \ln x + c + r x) \)

By \( \text{simp add: } r \)-def

Have \( r; r \in o(\lambda x. 1) \) unfolding \( r \)-def

Using tendsto-add[\( OF \) c tendsto-const[\( of - c \)] by \( \text{intro smalloI-tendsto\auto} \)

Define \( r' \) where \( r' = (\lambda x. \text{integral } \{ 2..x \} \ r) \)

Have \( \text{integrable-r: } r \text{ integrable-on } \{ x..y \} \)

If \( 2 \leq x \) for \( x \ y :: \text{real} \) using that unfolding \( r \)-def

By \( \text{intro integrable-diff integrable-primes-M} \)

(\( \text{auto intro!}: \text{integrable-continuous-real continuous-intros} \)

Hence integral: \( (r \text{ has-integral } r' x) \{ 2..x \} \) if \( x \geq 2 \) for \( x \)

By \( \text{auto simp: has-integral-iff } r' \)-def

Have \( r'; r' \in o(\lambda x. x) \) using \( \text{integrable-r unfolding } r' \)-def

By \( \text{intro integral-smallo[\( OF \) r]} \) (\( \text{auto simp: filterlim-ident} \)

Define \( C \) where \( C = 2 \ast (c + \ln 2 - 1) \)

Have \( \vartheta \sim \text{[at-top]} (\lambda x. x + (r x \ast x + C - r' x)) \)

Proof \( \text{intro asymp-equiv-refl-ev eventually-mono[\( OF \) eventually-gt-at-top]} \)

Fix \( x :: \text{real} \) assume \( x; x > 2 \)

Have \( M \text{ has-integral } ((x \ast \ln x - x + c \ast x) - (2 \ast \ln 2 - 2 + c \ast 2) + r') \)

\( \{ 2..x \} \)

Unfolding \( M \)-expand using \( x \)

By \( \text{intro has-integral-add[\( OF \) fundamental-theorem-of-calculus integral]} \)

(\( \text{auto simp \( flip \): has-field-derivative-iff has-vector-derivative} \)

\( \text{intro!}: \text{derivative-eq-intros continuous-intros} \)

From has-integral-unique[\( OF \) \( \vartheta \)-conv-\( M \)-integral this]

Show \( \vartheta x = x + (r x \ast x + C - r' x) \) using \( x \)

By \( \text{simp add: field-simps } M \text{-expand } C \)-def)

Qed

Also have \( (\lambda x. r x \ast x + C - r' x) \in o(\lambda x. x) \)

Proof \( \text{intro sum-in-smallo } r \)

Show \( (\lambda x. C) \in o(\lambda x. x) \) by \text{real-asymp}

Qed \( \text{insert landau-o.small-big-mult[\( OF \) r, of } \lambda x. x \) r', \text{simp-all} \)

Hence \( (\lambda x. x + (r x \ast x + C - r' x)) \sim \text{[at-top]} (\lambda x. x) \)

By \( \text{subst asymp-equiv-add-right} \) \auto

Finally show \( \text{thesis by } auto \)

102
The asymptotics of $\psi$ and $\pi$ follow as simple corollaries.

**corollary $\psi$-asymptotics:** $\psi \sim [\text{at-top}] (\lambda x. x)$

**proof** –

- define $r$ where $r = (\lambda x. \psi x - \vartheta x)$
- have $r \in O(\lambda x. \ln x * \sqrt{x})$
  - unfolding $r$-def by (fact $\psi$-minus-$\vartheta$-bigo)
- also have $(\lambda x :: \text{real}. \ln x * \sqrt{x}) \in o(\lambda x. x)$
  - by real-asym
- finally have $r$: $r \in o(\lambda x. x)$.

- have $(\lambda x. \vartheta x + r x) \sim [\text{at-top}] (\lambda x. x)$
  - using $\vartheta$-asymptotics $r$ by (subst asymp-equiv-add-right)
  - thus $\text{thesis}$ by (simp add: $r$-def)

**qed**

**corollary prime-number-theorem:** $\pi \sim [\text{at-top}] (\lambda x. x / \ln x)$

**proof** –

- have $(\lambda x. (\pi x - \vartheta x / \ln x) + ((\vartheta x - x) / \ln x)) \in o(\lambda x. x / \ln x)$
- proof (rule sum-in-smallo)
  - have $(\lambda x. \pi x - \vartheta x / \ln x) \in O(\lambda x. x / \ln x ^ 2)$
    - by (rule $\pi$-$\vartheta$-bigo)
  - also have $(\lambda x. x / \ln x ^ 2) \in o(\lambda x. x / \ln x :: \text{real})$
    - by real-asym
  - finally show $(\lambda x. \pi x - \vartheta x / \ln x) \in o(\lambda x. x / \ln x)$.

- next
  - have eventually $(\lambda x :: \text{real}. \ln x > 0)$ at-top by real-asym
  - hence eventually $(\lambda x :: \text{real}. \ln x \neq 0)$ at-top by eventually-elim auto
  - thus $(\lambda x. (\vartheta x - x) / \ln x) \in o(\lambda x. x / \ln x)$
    - by (intro landau-o-small.divide-right asymp-eqiv-imp-diff-smallo $\vartheta$-asymptotics)

**qed**

thus $\text{thesis}$ by (simp add: diff-divide-distrib asymp-equiv-altdef)

**qed**

**lemma asymp-equivD-strong:**

- assumes $f \sim [F] g$ eventually $(\lambda x. f x \neq 0 \lor g x \neq 0)$ $F$
- shows $((\lambda x. f x / g x) \longrightarrow 1) \ F$

**proof** –

- from assms(1) have $((\lambda x. \text{if } f x = 0 \land g x = 0 \text{ then } 1 \text{ else } f x / g x) \longrightarrow 1) \ F$
  - by (rule asymp-equivD)
  - also have $\text{thesis} \longleftrightarrow \text{thesis}$
    - by (intro filterlim-cong eventually-mono[OF assms(2)]) auto
  - finally show $\text{thesis}$.

**qed**

**corollary prime-number-theorem':** $((\lambda x. \pi x / (x / \ln x)) \longrightarrow 1)$ at-top

**using** prime-number-theorem
by (rule asymp-eqivD-strong[OF - eventually-mono[OF eventually-gt-at-top[of 1]]]) auto

corollary prime-number-theorem":
(\(\lambda x\). card \(\{ p.\ prime\ p \wedge real\ p \leq x\}\) \(\sim\)\[\text{at-top}\](\(\lambda x\). \(\lfloor x/\ln x \rfloor\))

proof –
  have \(\pi = (\lambda x.\ card \(\{ p.\ prime\ p \wedge real\ p \leq x\}\))\)
  by (intro ext) (simp add: \(\pi\)-def prime-sum-upto-def)
with prime-number-theorem show \(\text{thesis}\) by simp
qed

corollary prime-number-theorem"":
(\(\lambda n.\ card \(\{ p.\ prime\ p \wedge real\ p \leq n\}\) \(\sim\)\[\text{at-top}\](\(\lambda n.\ real\ n / \ln (\text{real}\ n))\)

proof –
  have \((\lambda n.\ card \(\{ p.\ prime\ p \wedge real\ p \leq n\}\) \(\sim\)\[\text{at-top}\](\(\lambda n.\ real\ n / \ln (\text{real}\ n))\))
  using prime-number-theorem"
  by (rule asymp-eqiv-compose') (simp add: filterlim-real-sequentially)
thus \(\text{thesis}\) by simp
qed

Finally, we show that \(p_n \sim n \ln n\), where \(p_n\) denotes the \(n\)-th prime number.
We again follow a proof by Apostol, first showing the intermediate result
\(\ln \pi(x) \sim x\):

lemma \(\ln\pi\)-asymptotics: \((\lambda x.\ \ln (\pi\ x)) \sim\)\[\text{at-top}\]\[\ln\]

proof –
  from prime-number-theorem' have \((\lambda x.\ \ln (\pi\ x) / (x / \ln x)) \longrightarrow\) \[\text{at-top}\].
  hence \((\lambda x.\ \ln (\pi\ x) / (x / \ln x)) \longrightarrow\) \(\ln 1\) \[\text{at-top}\]
  by (rule tendsto-ln) auto
  also have \(\text{th\ this} \longleftrightarrow ((\lambda x.\ \ln x * (\ln (\pi\ x) / \ln x + \ln (\ln x / \ln x - 1)) \longrightarrow\) \(0\) \[\text{at-top}\]
  by (intro filterlim-cong eventually-mono[OF eventually-gt-at-top[of 2]])
  (auto simp: ln-div field-simps ln-mult \(\pi\)-pos)
  finally have \((\lambda x::\text{real}.\ \ln x * ((\ln (\pi\ x) / \ln x + \ln (\ln x / \ln x - 1)) / \ln x) \longrightarrow\) \(0\) \[\text{at-top}\]
  by (rule real-tendsto-divide-at-top) real-asym
  also have \(\text{th\ this} \longleftrightarrow ((\lambda x.\ \ln (\pi\ x) / \ln x + (\ln (\ln x / \ln x - 1)) \longrightarrow\) \(0\) \[\text{at-top}\]
  by (intro filterlim-cong eventually-mono[OF eventually-gt-at-top[of 2]])
  (auto simp: field-simps)
  finally have \((\lambda x.\ \ln (\pi\ x) / \ln x + ((\ln (\ln x / \ln x - 1)) / \ln x - 1)) \longrightarrow\) \(0 - (\text{-}1)\) \[\text{at-top}\]
  by (rule tendsto-diff) real-asym
  hence \((\lambda x.\ \ln (\pi\ x) / \ln x) \longrightarrow\) \(1\) \[\text{at-top}\]
  by simp
  thus \((\lambda x.\ \ln (\pi\ x)) \sim\)\[\text{at-top}\]\[\ln\]
  by (rule asymp-eqivI')
qed
Combining this auxiliary result with the Prime Number Theorem, we obtain

\(\pi(x) \ln \pi(x) \sim x\), and with the substitution \(x := p_n:\)

\[ p_n \sim \pi(p_n) \ln \pi(p_n) \sim n \ln n. \]

corollary nth-prime-asymptotics: \((\lambda n. \text{real} \ (\text{nth-prime} \ n)) \sim[\text{at-top}] (\lambda n. \text{real} \ n * \ln \ (\text{real} \ n))\)

proof –

- have \((\lambda x. \pi \ x * \ln \ (\pi \ x)) \sim[\text{at-top}] (\lambda x. \ x / \ln x * \ln x)\)
  by (intro asymp-equiv-intros prime-number-theorem ln-\pi-asymptotics)

- also have \(\ldots \sim[\text{at-top}] (\lambda x. \ x)\)
  by (intro asymp-equiv-refl-ev eventually-mono[OF eventually-gt-at-top[of 1]])
  (auto simp: field-simps)

finally have \((\lambda x. \pi \ x * \ln \ (\pi \ x)) \sim[\text{at-top}] (\lambda x. \ x)\)
  by simp

hence \((\lambda n. \text{nth-prime} \ n) \sim[\text{at-top}] (\lambda n. \pi \ (\text{nth-prime} \ n) * \ln \ (\pi \ (\text{nth-prime} \ n)))\)
  by (rule asymp-equiv-symI[OF asymp-equiv-compose])
  (auto intro: filterlim-compose[OF filterlim-real-sequentially nth-prime-at-top])

- also have \(\ldots = (\lambda n. \text{real} \ (\text{Suc} \ n) * \ln \ (\text{real} \ (\text{Suc} \ n)))\)
  by (simp add: add-ac)

- also have \(\ldots \sim[\text{at-top}] (\lambda n. \text{real} \ n * \ln \ (\text{real} \ n))\)
  by real-asymp

finally show \(?thesis\).
qed

end

5 Mertens’ Theorems

theory Mertens-Theorems
imports
Prime-Counting-Functions
Stirling-Formula Stirling-Formula

begin

In this section, we will prove Mertens’ First and Second Theorem. These are weaker results than the Prime Number Theorem, and we will derive them without using it.

However, like Mertens himself, we will not only prove them asymptotically, but absolutely. This means that we will show that the remainder terms are not only “Big-O” of some bound, but we will give concrete (and reasonably tight) upper and lower bounds for them that hold on the entire domain. This makes the proofs a bit more tedious.
5.1 Absolute Bounds for Mertens’ First Theorem

We have already shown the asymptotic form of Mertens’ first theorem, i.e. \( \mathfrak{M}(n) = \ln n + O(1) \). We now want to obtain some absolute bounds on the \( O(1) \) remainder term using a more careful derivation than before.

The precise bounds we will show are \( \mathfrak{M}(n) - \ln n \in (-1 - \frac{9}{\pi^2}; \ln 4] \approx (-1.9119; 1.3863] \) for \( n \in \mathbb{N} \).

First, we need a simple lemma on the finiteness of exponents to consider in a sum of all prime powers up to a certain point:

**Lemma** exponents-le-finite:

- **assumes** \( p > (1 :: \text{nat}) \) \( k > 0 \)
- **shows** \( \text{finite} \{ \text{i. real } (p ^ (k * i + l)) \leq x \} \)
- **proof** (rule \text{finite-subset})
  - **show** \( \{ \text{i. real } (p ^ (k * i + l)) \leq x \} \subseteq \{ \text{..nat } |x| \} \)
  - **proof safe**
    - **fix** \( i \) **assume** \( i : \text{real } (p ^ (k * i + l)) \leq x \)
    - **have** \( i < 2 ^ i \) **by** (induction \( i \)) **auto**
    - **also from** assms **have** \( i \leq k * i + l \) **by** (cases \( k \)) **auto**
    - **hence** \( 2 ^ i \leq (2 ^ (k * i + l)) :: \text{nat} \)
      - **using** assms **by** (intro power-increasing) **auto**
      - **also have** \( \ldots \leq p ^ (k * i + l) \) **using** assms **by** (intro power-mono) **auto**
      - **also have** \( \text{real } \ldots \leq x \) **using** \( i \) **by** simp
      - **finally show** \( i \leq \text{nat } |x| \) **by** linarith
  - **qed**
  - **qed auto**

Next, we need the following bound on \( \zeta'(2) \):

**Lemma** deriv-zeta-2-bound: \( \text{Re } (\text{deriv zeta } 2) > -1 \)
- **proof**
  - **have** \( ((\lambda x :: \text{real. ln } (x + 3) * (x + 3) ^ (-2)) \text{ has-integral } (\ln 3 + 1) / 3) \)
    - **(interior \{0..\})
      - **using** \( \ln ^ {-\text{powr}} \text{-has-integral-at-top[of } 1 \text{ 0 3 } -2] \)
        **by** (simp add: interior-real-semiline powr-minus)
      - **hence** \( ((\lambda x :: \text{real. ln } (x + 3) * (x + 3) ^ (-2)) \text{ has-integral } (\ln 3 + 1) / 3) \)
        \{0..\}
      - **by** (subst (asm) \text{has-integral-interior}) **auto**
    - **also have** \( ?this \leftrightarrow ((\lambda x :: \text{real. ln } (x + 3) / (x + 3) ^ (-2)) \text{ has-integral } (\ln 3 + 1) / 3) \)
      \{0..\}
      **by** (intro has-integral-cong) **auto simp: powr-minus field-simps**
    - **finally have** int: \( \ldots \).\)
  - **have** \( \exp (1 / 2 :: \text{real}) ^ (-2) \leq 2 ^ (-2) \)
    **using** \( \exp ^ {-\text{le}} \) **by** (subst exp-double [symmetric]) **simp-all**
  - **hence** \( \exp ^ {-\text{half}}: \exp (1 / 2 :: \text{real}) \leq 2 \)
    **by** (rule power2-le-imp-le) **auto**
  - **have** mono: \( \ln x / x ^ (-2) \leq \ln y / y ^ (-2) \) \text{if } \( y \geq \exp (1/2) \) \( x \geq y \) \text{for } \( x \) \( y :: \text{real} \)
    **proof** (rule DERIV-nonpos-imp-nonincreasing[of - - \lambda x. ln x / x ^ (-2)])
  - **fix** \( t \) **assume** \( t \geq y \) \( t \leq x \)

106
have \(y > 0\) by (rule less-le-trans[OF that(1)]) auto
with \(t\) that have \(\ln t \geq \ln (\exp (1/2))\)
by (subst ln-le-cancel-iff) auto
hence \(\ln t \geq 1/2\) by (simp only: ln-exp)
from \(t\) we have \((\lambda x. \ln x / x ^ 2)\) has-field-derivative \((1 - 2 * \ln t) / (t ^ 3)\) (at \(t\))
by (auto intro!: derivative-eq-intros simp: eval-fds-deriv-zeta fds-nth-deriv scaleR-conv-of-real
simp del: of-nat-Suc)

have fds-converges (fds-deriv fds-zeta) (2 :: complex)
by (intro fds-converges-deriv) auto
hence \((\lambda n. \ln (\real (Suc n)) / (\of-nat (Suc n) ^ 2))\) sums deriv zeta 2
by (auto simp: fds-converges-altdef add-ac eval-fds-deriv-zeta fds-nth-deriv
scaleR-conv-of-real
simp del: of-nat-Suc)

note * = sums-split-initial-segment[OF sums-minus[OF sums-Re[OF OF this]], of 3]

have \((\lambda n. \ln (\real (n+4)) / (\real (n+4) ^ 2)\) sums \((-\Re (\text{deriv zeta} 2) - (\ln 2 / 4 + \ln 3 / 9))\)
using * by (simp add: eval-nat-numeral)

hence \(-\Re (\text{deriv zeta} 2) - (\ln 2 / 4 + \ln 3 / 9) =
\sum n. \ln (\real (Suc n) + 3) / (\real (Suc n) + 3) ^ 2\)
by (simp-all add: sums-iff algebra-simps)
also have \(\ldots \leq (\ln 3 + 1) / 3\) using int exp-half
by (intro decreasing-sum-le-integral divide-nonneg-pos mono) (auto simp: power-minus
field-simps)

finally have \(-\Re (\text{deriv zeta} 2) \leq (16 * \ln 3 + 9 * \ln 2 + 12) / 36\)
by (simp add: field-simps)
also have \(\ln 3 \leq (11 / 10 :: \real)\)
using ln-approx-bounds[of 3 2] by (simp add: power-nat-reduce numeral-2-eq-2)
hence \((16 * \ln 3 + 9 * \ln 2 + 12) / 36 \leq (16 * (11 / 10) + 9 * 25 / 36 + 12) / (36 :: \real)\)
using ln2-le-25-over-36 by (intro mono mult-left-mono divide-right-mono)
auto
also have \(\ldots < 1\) by simp
finally show \(\text{thesis}\) by simp

qed

Using the logarithmic derivative of Euler’s product formula for \(\zeta(s)\) at \(s = 2\) 
and the bound on \(\zeta'(2)\) we have just derived, we can obtain the bound 

\[
\sum_{p^i \leq x, i \geq 2} \frac{\ln p}{p^i} < \frac{9}{\pi^2} .
\]

lemma mertens-remainder-aux-bound:
fixes $x :: \mathbb{R}$
defines $R \equiv (\sum (p, i) \mid \text{prime } p \land i > 1 \land \mathbb{R} (p ^ i) \leq x, \ln (\mathbb{R} p) / p ^ i)$
shows $R < 9 / p i^2$

proof ~
define $S'$ where $S' = \{(p, i). \text{prime } p \land i > 1 \land \mathbb{R} (p ^ i) \leq x\}$
define $S''$ where $S'' = \{(p, i). \text{prime } p \land i > 1 \land \mathbb{R} (p ^ Suc i) \leq x\}$

have finite-row: finite $\{i. i > 1 \land \mathbb{R} (p ^ (i + k)) \leq x\}$ if $p: \text{prime } p$ for $p k$
proof (rule finite-subset)
  show $\{i. i > 1 \land \mathbb{R} (p ^ (i + k)) \leq x\} \subseteq \{\text{nats } [x]\}$
proof safe
  fix $i$ assume $i: i > 1 \land \mathbb{R} (p ^ (i + k)) \leq x$
  have $i < 2 ^ (i + k)$ by (induction $i$) auto
  also from $p$ have $\ldots \leq p ^ (i + k)$ by (intro power-mono) (auto dest: prime-gt-1-nat)
  also have $\ldots \leq x$ using $i$ by simp
  finally show $i \leq \text{nats } [x]$ by linarith
qed

have $S'' \subseteq S'$ unfolding $S''$-def $S'$-def
proof safe
  fix $p i$ assume $pi: \text{prime } p \mathbb{R} (p ^ Suc i) \leq x i > 1$
  have $\mathbb{R} (p ^ i) \leq \mathbb{R} (p ^ Suc i)$
    using $pi$ unfolding of-nat-le-iff by (intro power-increasing) (auto dest: prime-gt-1-nat)
  also have $\ldots \leq x$ by fact
  finally show $\mathbb{R} (p ^ i) \leq x$.
qed

have $S'$-alt: $S' = (\Sigma i: \{p. \text{prime } p \land \mathbb{R} p \leq x\}. \{i. i > 1 \land \mathbb{R} (p ^ i) \leq x\})$
unfolding $S'$-def
proof safe
  fix $p i$ assume prime $p \mathbb{R} (p ^ i) \leq x i > 1$
  hence $p ^ 1 \leq p ^ i$
    by (intro power-increasing) (auto dest: prime-gt-1-nat)
  also have $\ldots \leq x$ by fact
  finally show $\mathbb{R} p \leq x$ by simp
qed

have finite: finite $\{p. \text{prime } p \land \mathbb{R} p \leq x\}$
  by (rule finite-subset[\OF - finite-Nats-le-real[of x]]) (auto dest: prime-gt-0-nat)
have finite $S'$ unfolding $S'$-alt using finite-row[of - 0]
  by (intro finite-Sigma finite) auto

have $R \leq 3 / 2 * (\sum (p, i) \mid (p, i) \in S' \land \text{even } i. \ln (\mathbb{R} p) / p ^ i)$
proof ~
have $R = (\sum y \in \{0, 1\}: \sum z \mid z \in S' \land \text{snd } z \mod 2 = y. \ln (\mathbb{R} (\text{fst } z)) /$
real (fst z ^ snd z)
using (finite S') by (subst sum-group) (auto simp: case-prod-unfold R-def S'-def)
also have ... = (\sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ i)) +
(\sum (p,i) | (p, i) \in S' \land odd i. ln (real p) / real (p ^ i))
unfolding even-iff-mod-2-eq-zero odd-iff-mod-2-eq-one by (simp add: case-prod-unfold)
also have (\sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ Suc i)) =
(\sum (p,i) | (p, i) \in S'' \land even i. ln (real p) / real (p ^ Suc i))
by (intro sum.reindex-bij-witness[of - \lambda(p,i). (p, Suc i) \lambda(p,i). (p, i - 1)])
(auto simp: case-prod-unfold S'-def S''-def elim: oddE simp del: power-Suc)
also have ... \leq (\sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ Suc i))
using (S'' \subseteq S' unfolding case-prod-unfold
by (intro sum-mono2 divide-nonneg-pos ln-ge-zero finite-subset[OF - (finite S'')])
(auto simp: S'-def S''-def case-prod-unfold dest: prime-gt-0-nat simp del: power-Suc)
also have ... \leq (\sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ Suc i))
by (subt sum-distrib-left) (auto simp: case-prod-unfold)
also have (\sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ Suc i)) + ...
= \frac{3}{2} \sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ Suc i))
by simp
finally show ?thesis by simp
qed

also have (\sum (p,i) | (p, i) \in S' \land even i. ln (real p) / real (p ^ Suc i)) =
(\sum p | prime p \land real p \leq x. ln (real p) *
(\sum i | i > 0 \land even i \land real (p ^ i) \leq x. (1 / real p) ^ i))
unfolding sum-distrib-left
proof (subt sum.Sigma[OF - ballI])
fix p assume p: p \in \{ p. prime p \land real p \leq x \}
thus infinite { i. 0 < i \land even i \land real (p ^ i) \leq x }
by (intro infinite-subset[OF - exponents-le-finite[of 1 0 x]]) (auto dest: prime-gt-1-nat)
qed (auto intro!: sum.cong finite-subset[OF - finite-Nats-le-real[of x]]
dest: prime-gt-0-nat simp: S'-alt power-divide)
also have ... \leq (\sum p | prime p \land real p \leq x. ln (real p) / (real p ^ Suc 2 - 1))
proof (rule sum-mono)
fix p assume p: p \in \{ p. prime p \land real p \leq x \}
have p > 1 using p by (auto dest: prime-gt-1-nat)
have (\sum i | i > 0 \land even i \land real (p ^ Suc i) \leq x. (1 / real p) ^ Suc i) =
(\sum i | real (p ^ Suc i) \leq x. (1 / real p) ^ Suc i) / real p ^ Suc 2
(is - = ?S / -) unfolding sum-divide-distrib
by (rule sum.reindex-bij-witness[of - \lambda i. 2 * Suc i \lambda i. (i - 2) div 2])
(insert \{ p > 1 \}, auto simp: numeral-3-eq-3 power2-eq-square power-diff
algebra-simps elim!: evenE)
We now consider the equation
\[ \text{also have } S = \left( \sum_{i} \text{real } (p \cdot (2 + i + 2)) \right) \leq x \cdot (1 / \text{real } p^2) \]
by (\text{subst power-mult}) (\text{simp-all add: algebra-simps power-divide})
also have \(...\) \leq \left( \sum_{i} \text{real } (1 / \text{real } p^2) \right) \)
using (\text{exponent-s-letter-finite}) (\text{simp add: summable-geometric-iff})
also have \(...\) = \left( \sum_{i} \text{real } p^2 \right) / (\text{real } p^2 - 1)
using (\text{simp add: divide-simps})
finally have \(...\) \leq \left( \sum_{i} \text{real } p^2 \right) (\text{is } \text{lhs} \leq \text{rhs})
using (\text{simp add: divide-right-mono})
thus \text{ln } (\text{real } p) \leq \text{ln } (\text{real } p) / (\text{real } p^2 - 1)
using (\text{simp add: divide-simps})
\text{qed}
also have \(...\) = \left( \sum_{a} p \right) \text{prime } p \leq x \cdot \text{ln } (\text{real } p) / (\text{real } p^2 - 1))
using (\text{finite}) by (\text{intro infsum-finite}) (\text{symmetric})
also have \(...\) \leq \left( \sum_{a} p \right) \text{prime } p \cdot \text{ln } (\text{real } p) / (\text{real } p^2 - 1))
using (\text{eval-fds-logder-zeta-real[of 2]}) \text{finite}
by (\text{intro infsum-finite}) (\text{prime-neutral-left divide-nonneg-pos}) (\text{auto simp: dest: prime-ge-1-nat})
also have \(...\) = \text{Re } (\text{deriv zeta-of-real 2}) / (\text{zeta-of-real 2})
by (\text{subst eval-fds-logder-zeta-real}) \text{auto}
also have \(...\) = \text{Re } (\text{deriv zeta 2}) * (6 / pi^2)
by (\text{simp add: zeta-even-numeral})
also have \(...\) \leq 1 * (6 / pi^2)
using (\text{deriv-zeta-2-bound}) by (\text{intro mult-strict-right-mono}) \text{auto}
also have 3 / 2 * \text{...} = 9 / pi^2 by \text{simp}
finally show \text{thesis} by \text{simp}
\text{qed}

We now consider the equation
\[ \ln(n!) = \sum_{k \leq n} \Lambda(k) \left\lfloor \frac{n}{k} \right\rfloor \]
and estimate both sides in different ways. The left-hand-side can be estimated using Stirling’s formula, and we can simplify the right-hand side to
\[ \sum_{k \leq n} \Lambda(k) \left\lfloor \frac{n}{k} \right\rfloor = \sum_{p^i \leq x, i \geq 1} \ln p \left\lfloor \frac{n}{p^i} \right\rfloor \]
and then split the sum into those \( p^i \) with \( i = 1 \) and those with \( i \geq 2 \). Applying the bound we have just shown and some more routine estimates, we obtain the following reasonably strong version of Mertens’ First Theorem on the naturals:
\[ \mathcal{M}(n) - \ln(n) \in (-1 - \frac{9}{72}; \ln 4) \]

\text{theorem mertens-bound-strong:}
\text{fixes } n :: \text{nat} \text{ assumes } n : n > 0
\text{110}
shows \( M_n - \ln n \in \{-1 - 9 / \pi^2 < \ldots \ln 4\} \)

proof (cases \( n \geq 3 \))

case False

with \( n \) consider \( n = 1 \mid n = 2 \) by force

thus ?thesis

proof

\begin{align*}
\text{assume} \ [\text{simpl}] : n = 1 \\
\text{have} \ (-1 + (-9 / \pi^2) < 0 \text{ by intro add-neg-neg divide-neg-pos}) \\
\text{auto} \text{ by simp}
\end{align*}

next

\begin{align*}
\text{assume} \ [\text{simpl}] : n = 2 \\
\text{have eq} : M_n - \ln n = -\ln 2 / 2 \text{ by simp add: eval-M} \\
\text{have} \ (-1 - 9 / \pi^2 + \ln 2 / 2 \leq -1 - 9 / 4 ^ 2 + 25 / 36 / 2 \\
\text{using pi-less-4 ln2-le-25-over-36 by intro diff-mono add-mono divide-left-mono divide-right-mono power-mono}) \\
\text{auto} \text{ by simp}
\end{align*}

\begin{align*}
\text{also have} \ldots < 0 \text{ by simp} \\
\text{finally have} \ -\ln 2 / 2 > -1 - 9 / \pi^2 \text{ by simp} \\
\text{moreover} \{ \\
\text{have} -\ln 2 / 2 \leq (0 :: \text{real}) \text{ by intro divide-nonpos-pos}) \text{ auto} \\
\text{also have} \ldots \leq \ln 4 \text{ by simp} \\
\text{finally have} \ -\ln 2 / 2 \leq \ln (4 :: \text{real}) \text{ by simp} \\
\}
\text{ultimately show} \ ?thesis \text{ unfolding eq by simp}
\end{align*}

qed

\begin{align*}
\text{next}
\text{case True}
\text{hence} \ n : n \geq 3 \text{ by simp}
\text{have finite: finite \{(p, i). prime p \land i \geq 1 \land p ^ i \leq n \}}
\text{proof}
\text{rule finite-subset}
\text{ show \{(p, i). prime p \land i \geq 1 \land p ^ i \leq n \}
\subseteq \{/ \text{nat \{root 1 (real n)\}}\} \times \{/ \text{nat \{log 2 (real n)\}}\}
\text{using primepows-le-subset[of real n 1] n unfolding of-nat-le-iff by auto}
\text{qed auto}
\end{align*}

\begin{align*}
\text{define} \ r \text{ where} \ r = \text{prime-sum-upto} (\lambda p. \ln (\text{real } p) \ast \text{frac (real } n \text{ / real } p)) \text{ n} \\
\text{define} \ R \text{ where} \ R = (\sum (p, i) \mid \text{prime } p \land i > 1 \land p ^ i \leq n. \ln (\text{real } p) \ast \text{real } (n \text{ div } (p ^ i))) \\
\text{define} \ R' \text{ where} \ R' = (\sum (p, i) \mid \text{prime } p \land i > 1 \land p ^ i \leq n. \ln (\text{real } p) / p ^ i)
\text{have [simp]:} \ln (4 :: \text{real}) = 2 * \ln 2 \\
\text{using ln-realpow[of 2 2] by simp}
\text{from pi-less-4 have} \ \ln pi \leq \ln 4 \text{ by (subst ln-le-cancel-iff) auto}
\text{also have} \ldots = 2 * \ln 2 \text{ by simp}
\text{also have} \ldots \leq 2 * (25 / 36) \text{ by (intro mult-left-mono ln2-le-25-over-36) auto}
\text{finally have ln-pi:} \ln pi \leq 25 / 18 \text{ by simp}
\text{have ln 3 \leq ln (4 :: nat) by (subst ln-le-cancel-iff) auto}
\end{align*}
also have \ldots = 2 * \ln 2 \textbf{ by simp}
also have \ldots \leq 2 * (25 / 36) \textbf{ by (intro mult-left-mono ln2-le-25-over-36) auto}
finally have \( \ln 3: \ln (3::\text{real}) \leq 25 / 18 \textbf{ by simp} \)

have \( R / n = (\sum (p,i) \mid \text{prime } p \land i > 1 \land p ^ \wedge i \leq n, \ln (\text{real } p) \ast (\text{real } (n \div (p ^ \wedge i)) / n)) \)
\hspace{1em} by (\text{simp add: R-def sum-divide-distrib field-simps case-prod-unfold})
also have \ldots \leq (\sum (p,i) \mid \text{prime } p \land i > 1 \land p ^ \wedge i \leq n, \ln (\text{real } p) \ast (1 / p ^ \wedge i))
\hspace{1em} \textbf{unfolding R'-def case-prod-unfold using } n
\hspace{1em} by (\text{intro sum-mono mult-left-mono}) (\text{auto simp: field-simps real-of-nat-div dest: prime-gt-0-nat})
also have \ldots = R' \textbf{ by (simp add: R'-def)}
also have R' < 9 / \pi^2
\hspace{1em} \textbf{unfolding R'-def using mertens-remainder-aux-bound[of n] by simp}
finally have R / n < 9 / \pi^2 .
moreover have R \geq 0
\hspace{1em} \textbf{unfolding R-def by (intro sum-nonneg mult-nonneg-nonneg) (auto dest: prime-gt-0-nat)}
ultimately have R-bounds: R / n \in \{0..<9 / \pi^2\} \textbf{ by simp}

have \( \ln (\text{fact } n :: \text{real}) \leq \ln (2 * \pi * n) / 2 + n * \ln n - n + 1 / (12 * n) \)
\hspace{1em} \textbf{using ln-fact-bounds(2)[of n] n by simp}
also have \ldots / n = \ln n = -1 + (\ln 2 + \ln \pi) / (2 * n) + (\ln n / n) / 2 + 1 / (12 * \text{real } n ^ \wedge 2)
\hspace{1em} \textbf{using } n \textbf{ by (simp add: power2-eq-square field-simps ln-mult)}
also have \ldots \leq -1 + (\ln 2 + \ln \pi) / (2 * 3) + (\ln 3 / 3) / 2 + 1 / (12 * 3^2)
\hspace{1em} \textbf{using exp-le n pi-gt3}
\hspace{1em} \textbf{by (intro add-mono divide-right-mono divide-left-mono mult-mono}
\hspace{1em} \multimap pos-pos ln-x-over-x-mono power-mono) \textbf{ auto}
also have \ldots \leq -1 + (25 / 36 + 25 / 18) / (2 * 3) + (25 / 18 / 3) / 2 + 1 / (12 * 3^2)
\hspace{1em} \textbf{using ln-pi ln2-le-25-over-36 ln3 by (intro add-mono divide-left-mono divide-right-mono) auto}
also have \ldots \leq 0 \textbf{ by simp}
finally have \( \ln n = \ln (\text{fact } n) / n \geq 0 \textbf{ using } n \textbf{ by (simp add: divide-right-mono)}
have \( -\ln (\text{fact } n) \leq -\ln (2 * \pi * n) / 2 - n * \ln n + n \)
\hspace{1em} \textbf{using ln-fact-bounds(1)[of n] n by simp}
also have \ln n + \ldots / n = -\ln (2 * \pi) / (2 * n) - (\ln n / n) / 2 + 1
\hspace{1em} \textbf{using } n \textbf{ by (simp add: field-simps ln-mult)}
also have \ldots \leq 0 - 0 + 1
\hspace{1em} \textbf{using pi-gt3 n by (intro add-mono diff-mono) auto}
finally have upper: \ln n - \ln (\text{fact } n) / n \leq 1
\hspace{1em} \textbf{using } n \textbf{ by (simp add: divide-right-mono)}
\hspace{1em} \textbf{with } \ln n - \ln (\text{fact } n) / n \geq 0 \textbf{ have fact-bounds: ln n - ln (fact n) / n \in } \{0..1\} \textbf{ by simp}

have \( r \leq \text{prime-sum-upto} (\lambda p. \ln p * 1) \textbf{ n}
\hspace{1em} \textbf{using less-imp-le[OF frac-lt-1]} \textbf{ unfolding r-def \vartheta-def prime-sum-upto-def} \)
As a simple corollary, we obtain a similar bound on the reals.

**Lemma mertens-bound-real-strong:**
fixes $x :: real$
assumes $x :: x \geq 1$
shows $\mathbb{M} \ x - \ln x \in \{-1 - 9 / \pi^2 < \ln 4\}$
proof
  have $\mathbb{M} \ x - \ln x \leq \mathbb{M} \ (\text{real} \ (\text{nat} \ |x|)) - \ln \ (\text{real} \ (\text{nat} \ |x|))$
    using assms by simp
qed
also have \(\ldots \leq \ln 4\)
using mertens-bound-strong[of nat \([x]\)] assms by simp
finally have \(\mathbb{M} x - \ln x \leq \ln 4\).

from assms have \(\text{pos: real-of-int} \ [x] \neq 0\) by linarith
have \(\frac{x}{\ln 2} / \text{real} (\text{nat} \ [x]) \geq 0\)
using assms by (intro divide-nonneg-pos) auto
moreover have \(\frac{x}{\ln 4} / \text{real} (\text{nat} \ [x]) \leq 1 / 1\)
using assms frac-le-1[of \(x\)] by (intro frac-le) auto
ultimately have \(\frac{x}{\ln 4} / \text{real} (\text{nat} \ [x]) \in \{0..1\}\) by auto
have \(\ln x - \ln (\text{real} (\text{nat} \ [x])) = \ln (x / \text{real} (\text{nat} \ [x]))\)
using assms by (subst ln-div) auto
also have \(\frac{x}{\ln 4} / \text{real} (\text{nat} \ [x]) = 1 + \frac{x}{\ln 4} / \text{real} (\text{nat} \ [x])\)
using assms pos by (simp add: frac-def field-simps)
finally have \(\mathbb{M} x - \ln x > -1 - 9/\pi^2 - \ln (1 + \frac{x}{\ln 4} / \text{real} (\text{nat} \ [x]))\)
using mertens-bound-strong[of nat \([x]\)] \(x\) by simp
with \(\mathbb{M} x - \ln x \leq \ln 4\) show \(\thefalse\) by simp
qed

We weaken this estimate a bit to obtain nicer bounds:

lemma mertens-bound-real\(^4\):
fixes \(x :: \text{real}\) assumes \(x: x \geq 1\)
shows \(\mathbb{M} x - \ln x \in\{-2..<25/18\}\)
proof –
have \(\mathbb{M} x - \ln x \leq \ln 4\)
using mertens-bound-real-strong[of \(x\)] \(x\) by simp
also have \(\ldots \leq 25 / 18\)
using ln-realpow[of 2 \(2\)] ln2-le-25-over-36 by simp
finally have \(\mathbb{M} x - \ln x \leq 25 / 18\).

have \(\ln 2: \ln (2 :: \text{real}) \in \{2/3..25/36\}\)
using ln-approx-bounds[of 2 \(1\)] by (simp add: eval-nat-numeral)
have \(\ln 3: \ln (3 :: \text{real}) \in \{1..10/9\}\)
using ln-approx-bounds[of 3 \(1\)] by (simp add: eval-nat-numeral)
have \(\ln 5: \ln (5 :: \text{real}) \in \{4/3..76/45\}\)
using ln-approx-bounds[of 5 \(1\)] by (simp add: eval-nat-numeral)
have \(\ln 7: \ln (7 :: \text{real}) \in \{3/2..15/7\}\)
using ln-approx-bounds[of 7 \(1\)] by (simp add: eval-nat-numeral)
have \(\ln 11: \ln (11 :: \text{real}) \in \{5/3..290/99\}\)
using ln-approx-bounds[of 11 \(1\)] by (simp add: eval-nat-numeral)

— Choosing the lower bound -2 is somewhat arbitrary here; it is a trade-off between getting a reasonably tight bound and having to make lots of case distinctions. To get \(-2\) as a lower bound, we have to show the cases up to \(x = 11\) by case distinction,

have \(\mathbb{M} x - \ln x > -2\)
proof (cases \(x \geq 11\))
case False
hence \(x \in \{1..<2\} \lor x \in \{2..<3\} \lor x \in \{3..<5\} \lor x \in \{5..<7\} \lor x \in \ldots\)
\{7..<11\}

using \(x\) by force

thus \(?thesis\)

proof (elim disjE)

assume \(x: x \in \{1..<2\}\)

hence \(\ln x - M x \leq \ln 2 - 0\)

by (intro diff-mono) auto

also have \(\ldots < 2\) using \(\ln 2\)

by simp

finally show \(?thesis\) by simp

next

assume \(x: x \in \{2..<3\}\)

hence [simp]: \([x] = 2\) by (intro floor-unique) auto

from \(x\) have \(\ln x - M x \leq \ln 3 - \ln 2 / 2\)

by (intro diff-mono) (auto simp: eval-M)

also have \(\ldots = \ln (9/2) / 2\) using \(\ln\)-realpow\([of 3 2]\) by (simp add: ln-div)

also have \(\ldots < 2\) using \(\ln\)-approx-bounds\([of 9/2 1]\) by (simp add: eval-nat-numeral)

finally show \(?thesis\) by simp

next

assume \(x: x \in \{3..<5\}\)

hence \(M 3 = M x\)

unfolding primes-M-def

by (intro prime-sum-upto-eqI\([where a' = 3 \text{ and } b' = 4]\))

(auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)

also have \(M 3 = \ln 2 / 2 + \ln 3 / 3\)

by (simp add: eval-M eval-nat-numeral mark-out-code)

finally have [simp]: \(M x = \ln 2 / 2 + \ln 3 / 3 \ldots\)

from \(x\) have \(\ln x - M x \leq \ln 5 - (\ln 2 / 2 + \ln 3 / 3)\)

by (intro diff-mono) auto

also have \(\ldots < 2\) using \(\ln\)-realpow\([of 5\ldots]\) by simp

finally show \(?thesis\) by simp

next

assume \(x: x \in \{5..<7\}\)

hence \(M 5 = M x\)

unfolding primes-M-def

by (intro prime-sum-upto-eqI\([where a' = 5 \text{ and } b' = 6]\))

(auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)

also have \(M 5 = \ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5\)

by (simp add: eval-M eval-nat-numeral mark-out-code)

finally have [simp]: \(M x = \ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5 \ldots\)

from \(x\) have \(\ln x - M x \leq \ln 7 - (\ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5)\)

by (intro diff-mono) auto

also have \(\ldots < 2\) using \(\ln\)-realpow\([of 7\ldots]\) by simp

finally show \(?thesis\) by simp

next

assume \(x: x \in \{7..<11\}\)

hence \(M 7 = M x\)

unfolding primes-M-def

by (intro prime-sum-upto-eqI\([where a' = 7 \text{ and } b' = 10]\))
also have \( M \approx 0.261497 \) is the Meissel–Mertens constant.

We define the constant in the following way:

definition meissel-mertens where
meissel-mertens = 1 - ln (ln 2) + integral \( \{2..\} \) (\( \lambda t. (M t - ln t) / (t * ln t ^ 2) \))
We will require the value of the integral \( \int_a^\infty \frac{t}{\ln^2 t} \, dt = \frac{1}{\ln a} \) as an upper bound on the remainder term:

**lemma** integral-one-over-x-ln-x-squared:

assumes \( a : (\mathbb{R} : real) > 1 \)

shows set-integrable borel \( \{a < ..\} (\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2)) \) (is \( ?\text{th}\text{I} \))

and set-lebesgue-integral borel \( \{a < ..\} (\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2)) = 1 / \ln a \) (is \( ?\text{th}\text{I} \))

and \((\lambda t. 1 / (t * (\ln t)^2))\) has-integral \( 1 / \ln a \) \( \{a < ..\} \) (is \( ?\text{th}\text{I} \))

**proof**

have cont: isCont \((\lambda t. 1 / (t * (\ln t)^2)) x \) if \( x > a \) for \( x \)

using that \( a \) by \((\text{auto intro!: continuous-intros})\)

have deriv: \((\lambda x. -1 / \ln x)\) has-real-derivative \( 1 / (x * (\ln x)^2)\) \((\text{at} x)\) if \( x > a \) for \( x \)

using that \( a \) by \((\text{auto intro!: derivative-eq-intros simp: power2-eq-square field-simps})\)

have lim1: \(((\lambda x. -1 / \ln x) o \text{real-of-ereal}) \longrightarrow -(1 / \ln a)\) \((\text{at-right (ereal a)})\)

unfoldingereal-tendsto-simps using \( a \) by \((\text{real-asym simp: field-simps})\)

have lim2: \(((\lambda x. -1 / \ln x) o \text{real-of-ereal}) \longrightarrow 0\) \((\text{at-left \infty})\)

unfoldingereal-tendsto-simps using \( a \) by \((\text{real-asym simp: field-simps})\)

have set-integrable borel \((\text{einterval a \infty}) \) \((\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2))\)

by \((\text{rule interval-integral-FTC-nonneg[OF - deriv cont - lim1 lim2]})(\text{use a in auto})\)

thus \( ?\text{th}\text{I} \) by simp

have interval-lebesgue-integral borel \((\text{ereal a}) \infty \) \((\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2))\) = \(-1 \) \((\text{at-right \infty})\)

by \((\text{rule interval-FTC-tNonneg[OF - deriv cont - lim1 lim2]})(\text{use a in auto})\)

thus \( ?\text{th}\text{I} \) by simp add: interval-integral-to-infinity-eq

have \((\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2))\) has-integral

set-lebesgue-integral lebesgue \( \{a < ..\} \) \((\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2))\) \( \{a < ..\} \)

using \( ?\text{th}\text{I} \) by \((\text{intro has-integral-set-lebesgue})\)

(auto simp: set-integrable-def integrable-completion)

also have set-lebesgue-integral lebesgue \( \{a < ..\} \) \((\lambda t. 1 / (t * \ln t \ ^ {\cdot} 2)) = 1 / \ln a\)

using \( ?\text{th}\text{I} \) unfolding set-lebesgue-integral-def by \((\text{subst integral-completion})\)

auto

finally show \( ?\text{th}\text{I} \).

qed

We show that the integral in our definition of the Meissel–Mertens constant is well-defined and give an upper bound for its tails:

**lemma**

assumes \( a > (1 : \mathbb{R}) \)

defines \( r \equiv (\lambda t. (\text{ereal t} - \ln t) / (t * \ln t \ ^ {\cdot} 2))\)

shows integrable-meissel-mertens: set-integrable borel \( \{a < ..\} \) \( r \)

and meissel-mertens-integral-le: norm (integral \( \{a < ..\} \) \( r \)) \( \leq 2 / \ln a\)

**proof** –
have *: ((∀t. 2 * (1 / (t * ln t ^ 2)))) has-integral 2 * (1 / ln a) {a<..}
using assms by (intro has-integral-mult-right integral-one-over-x-ln-x-squared)

show set-integrable borel {a<..} r unfolding set-integrable-def
proof (rule Bochner-Integration.integrable-bound[OF - _ AE-I2])
  have integrable borel (λt::real. indicator {a<..} t * (2 * (1 / (t * ln t ^ 2))))
    using integrable-mult-right[of 2,
      OF integral-one-over-x-ln-x-squared(1)[of a, unfolded set-integrable-def]]
  thus (λt::real. indicator {a<..} t * (2 / (x * ln x ^ 2)))
    unfolding norm-scaleR norm-mult r-def norm-divide using mertens-first-theorem[of x] assms
      by (intro mult-mono frac-le divide-nonneg-pos) (auto simp: indicator-def)
    qed (auto simp: indicator-def)
    qed (auto simp: r-def)
  hence r integrable-on {a<..}
    by (simp add: set-borel-integral-eq-integral(1))
  hence norm (integral {a<..} r) ≤ integral {a<..} (λx. 2 * (1 / (x * ln x ^ 2)))
    using * by (simp add: has-integral-iff)
  fix x assume x ∈ {a<..}
  hence norm (r x) ≤ 2 / (x * (ln x) ^ 2)
    unfolding r-def norm-divide using mertens-first-theorem[of x] assms
      by (intro mult-mono frac-le divide-nonneg-pos) (auto simp: indicator-def)
    thus norm (r x) ≤ 2 * (1 / (x * ln x ^ 2)) by simp
  qed
  also have ... = 2 / ln a
    using * by (simp add: has-integral-iff)
  finally show norm (integral {a<..} r) ≤ 2 / ln a .
  qed

lemma integrable-on-meissel-mertens:
assumes A ⊆ {1..} Inf A > 1 A ∈ sets borel
shows (λt. (2π t − ln t) / (t * ln t ^ 2)) integrable-on A
proof –
  from assms obtain x where x: 1 < x x < Inf A
    using dense by blast
  from assms have bdd-below A by (intro bdd-below[OF - _]) auto
  hence A ⊆ {Inf A..} by (auto simp: cInf-lower)
  also have ... ⊆ {x<..} using x by auto

qed
Finally have \( \ast : A \subseteq \{ x < \} \).

have set-integrable lborel A (\( \lambda t. (\Re t - \ln t) / (t * \ln t ^{-2}) \))
by (rule set-integrable-subset[OF integrable-meissel-mertens[of x]]) (use \( x * \) 
assms in auto)
thus \( \ast \theta sis \) by (simp add: set-borel-integral-eq-integral(1))
qed

lemma meissel-mertens-bounds: \( |\text{meissel-mertens} - 1 + \ln (\ln 2)| \leq 2 / \ln 2 \)
proof –
have \( \ast : \{ 2.. \} = \{ 2:<.. \} = \{ 2::real \} \) by auto
also have negligible \( \ldots \) by simp
finally have integral \( \{ 2.. \} (\lambda t. (\Re t - \ln t) / (t * (\ln t^2))) = \)
algebraic \( \{ 2:<.. \} (\lambda t. (\Re t - \ln t) / (t * (\ln t^2))) \)
by (intro sym[OF integral-subset-negligible]) auto
also have \( \text{norm} \ldots \leq 2 / \ln 2 \)
by (rule meissel-mertens-integral-lc) auto
finally show \( \text{meissel-mertens} - 1 + \ln (\ln 2) \leq 2 / \ln 2 \)
by (simp add: meissel-mertens-def)
qed

Finally, obtaining Mertens’ second theorem from the first one is nothing but
a routine summation by parts, followed by a use of the above bound:

theorem mertens-second-theorem:
defines \( f \equiv \text{prime-sum-upto} (\lambda p. 1 / p) \)
shows \( \forall x. x \geq 2 \implies |f x - \ln (\ln x) - \text{meissel-mertens}| \leq 4 / \ln x \)
and \( (\lambda x. f x - \ln (\ln x) - \text{meissel-mertens}) \in O(\lambda x. 1 / \ln x) \)
proof –
define \( r \) where \( r = (\lambda t. (\Re t - \ln t) / (t * \ln t ^{-2})) \)
\{
fix \( x :: \text{real} \) assume \( x : x > 2 \)
have \( (\lambda t. \Re t * (1 / (t * \ln t ^{-2}))) \) has-integral \( \Re x * (1 / \ln x) - \Re 2 * (1 / \ln 2) \)
\( - (\sum n \in \text{real } -' \{ 2:<.. \} \text{ ind prime } n * (\ln n / \text{real } n) * (1 / \ln n)) \)
\( \{ 2.. \} \)
unfolding primes-M-def prime-sum-upto-altdef1 using \( x \)
by (intro partial-summation-strong[of \( \{ \}\) ])
(auto intro!: continuous-intros derivative-eq-intros simp: power2-eq-square
simp flip: has-field-derivative-iff-has-vector-derivative)
also have \( \Re x * (1 / \ln x) - \Re 2 * (1 / \ln 2) - (\sum n \in \text{real } -' \{ 2:<.. \} \text{ ind prime } n * (\ln n / \text{real } n) * (1 / \ln n)) = \)
\( \Re x / \ln x - (\sum n \in \text{insert } 2 \text{ real } -' \{ 2:<.. \} \text{ ind prime } n * (\ln n / \text{real } n) / (1 / \ln n)) \)
(is - = - = \( S \))
by (subst sum.insert)
(auto simp: primes-M-def finite-vimage-real-of-nat-greaterThanAtMost
eval-prime-sum-upto)
also have \( \text{\( S \) = \( f \) } \)
unfolding f-def prime-sum-upto-altdef1 sum-upto-def using \( x \)
by (intro sum.mono-neutral-cong-left) (auto simp: not-less numeral-2-eq-2 le-Suc-eq)
finally have \((\lambda t. -\Re t / (t * \ln t ^ 2))\) has-integral \((\Re x / \ln x - f x)\) \{2..\} by simp
from has-integral-neg[OF this] have \((\lambda t. \Re t / (t * \ln t ^ 2))\) has-integral \((f x - \Re x / \ln x)\) \{2..\} by simp
hence \((\lambda t. \Re t / (t * \ln t ^ 2) - 1 / (t * \ln t))\) has-integral \((f x - \Re x / \ln x - (\ln (\ln x) - \ln (\ln 2)))\) \{2..\} using x
by (intro has-integral-diff fundamental-theorem-of-calculus)
(auto simp flip: has-field-derivative-iff-has-vector-derivative intro!: derivative-eq-intros)
also have \(?this \iff (r has-integral (f x - \Re x / \ln x - (\ln (\ln x) - \ln (\ln 2)))\}) \{2..\}
by (intro has-integral-cong) (auto simp: r-def field-simps power2-eq-square)
finally have \ldots.
}
note integral = this

define R\_M where \(R_M = (\lambda x. \Re x - \ln x)\)
have \(\Re; \Re x = \ln x + R_M x\) for x by (simp add: R\_M-def)
define I where \(I = (\lambda x. \text{integral} \{x,\} r)\)
define C where \(C = (1 - \ln (\ln 2) + I 2)\)
have C-altdef: \(C = \text{meissel-mertens}\)
  by (simp add: I-def r-def C-def meissel-mertens-def)

show bound: \(|f x - \ln (\ln x) - \text{meissel-mertens}| \leq 4 / \ln x\) if x: x \geq 2 for x
proof (cases x = 2)
case True
hence \(|f x - \ln (\ln x) - \text{meissel-mertens}| = 1 / 2 - \ln (\ln 2) - \text{meissel-mertens}|\)
  by (simp add: f-def ecal-prime-sum-upto)
also have \ldots \leq 2 / \ln 2 + 1 / 2
  using meissel-mertens-bounds by linarith
also have \ldots \leq 2 / \ln 2 + 2 / \ln 2 using ln2-le-25-over-36
  by (intro add-mono divide-left-mono) auto
finally show \?thesis using True by simp
next
case False
hence x: x \geq 2 using x by simp
  have integral \{2..\} r + I x = integral \{(2..) \cup \{x,\}\} r unfolding I-def
  r-def using x
  by (intro integral-Un [symmetric] integrable-on-meissel-mertens) (auto simp:
    max-def r-def)
also have \{2..\} \cup \{x,\} = \{2..\} using x by auto
finally have *: integral \{2..\} r = I 2 - I x unfolding I-def by simp
have eq: f x - \ln (\ln x) - C = R_M x / \ln x - I x
  using integral[OF x] x by (auto simp: C-def field-simps \Re has-integral-iff *)
also have \ldots \leq |R_M x / \ln x| + norm (I x)
  unfolding real-norm-def by (rule abs-triangle-ineq4)
also have $|R_{\Re} x / \ln x| \leq 2 / |\ln x|

unfolding $R_{\Re}-\text{def abs-divide}$ using mertens-first-theorem[of x] x
by (intro divide-right-mono) auto
also have $\{x..\} - \{x<..\} = \{x\}$ and $\{x<..\} \subseteq \{x..\}$ by auto
hence $I x = \text{integral} \{x<..\} r$ unfolding $I$-def
by (intro integral-subset-negligible [symmetric]) simp-all
also have norm $\ldots \leq 2 / \ln x$
using meissel-mertens-integral-le[of x] x by (simp add: r-def)
finally show $|f x - \ln (\ln x) - \text{meissel-mertens}| \leq 4 / \ln x$
using x by (simp add: C-altdef)

qed

have $(\lambda x. f x - \ln (\ln x) - C) \in O(\lambda x. 1 / \ln x)$
proof (intro \text{landau-o\_bigI}[of 4] eventually-mono[OF eventually-ge-at-top[of 2]])
fix $x :: \text{real}$ assume $x : x \geq 2$
with bound[OF x] show norm $(f x - \ln (\ln x) - C) \leq 4 * \text{norm} (1 / \ln x)$
by (simp add: C-altdef)

qed (auto intro!: add-pos-nonneg)

thus $(\lambda x. f x - \ln (\ln x) - \text{meissel-mertens}) \in O(\lambda x. 1 / \ln x)$
by (simp add: C-altdef)

qed

corollary prime-harmonic-asymptotic-equivalence: prime-sum-upto $(\lambda p. 1 / p) \sim [\text{at-top}]
(\lambda x. \ln (\ln x))$
proof -
define $f$ where $f = \text{prime-sum-upto} (\lambda p. 1 / p)$

have $(\lambda x. f x - \ln (\ln x) - \text{meissel-mertens} + \text{meissel-mertens}) \in o(\lambda x. \ln (\ln x))$
unfolding $f$-def
by (rule sum-in-small-o[OF \text{landau-o\_big-small-trans}[OF mertens-second-theorem(2)]])

real-asymptotic+

hence $(\lambda x. f x - \ln (\ln x)) \in o(\lambda x. \ln (\ln x))$
by simp
thus $?\text{thesis}$ unfolding $f$-def
by (rule smallo-imp-asymptotic-equivalence)

qed

As a corollary, we get the divergence of the prime harmonic series.
corollary prime-harmonic-diverges: filterlim $(\lambda p. 1 / p) \text{ at-top}$
at-top
using \text{asymptotic-equivalence}[OF prime-harmonic-asymptotic-equivalence]
by (rule asymptotic-equivalence-at-top-transfer) real-asymptotic

end

121
References


