The Prime Number Theorem

Manuel Eberl and Larry Paulson

May 26, 2024

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 Avigad 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 37

3 Prime-Counting Functions 52
   3.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
   3.2 Basic properties . . . . . . . . . . . . . . . . . . . . . . . . . . 55
   3.3 The \( n \)-th prime number . . . . . . . . . . . . . . . . . . . 57
   3.4 Relations between different prime-counting functions . . . . . 62
   3.5 Bounds . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
   3.6 Equivalence of various forms of the Prime Number Theorem . 74
   3.7 The asymptotic form of Mertens’ First Theorem . . . . . . . 79

4 The Prime Number Theorem 84
   4.1 Constructing Newman’s function . . . . . . . . . . . . . . . . . . 84
   4.2 The asymptotic expansion of \( \mathfrak{M} \) . . . . . . . . . . . . . 93
   4.3 The asymptotics of the prime-counting functions . . . . . . . 99

5 Mertens’ Theorems 101
   5.1 Absolute Bounds for Mertens’ First Theorem . . . . . . . . . . 102
   5.2 Mertens’ Second Theorem . . . . . . . . . . . . . . . . . . . . . 113

6 Acknowledgements 118
1 Auxiliary material

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

Conflicting notation from HOL−Analysis.Infinite-Sum

no-notation Infinite-Sum.abs-summable-on (infix abs'-summable'-on 46)

lemma homotopic-loopsI:
  fixes h :: real × real ⇒ -
  assumes continuous-on ({{0..1} × {0..1}}) h
            h ' ({{0..1} × {0..1}}) ⊆ s
  assumes ⋀x. x ∈ {0..1} ⇒ h (0, x) = p x
  assumes ⋀x. x ∈ {0..1} ⇒ h (1, x) = q x
  assumes ⋀x. x ∈ {0..1} ⇒ pathfinish (h ⪫ Pair x) = pathstart (h ⪫ Pair x)
  shows homotopic-loops s p q
  using assms unfolding homotopic-loops by (intro exI[of - h]) auto

lemma homotopic-pathsI:
  fixes h :: real × real ⇒ -
  assumes continuous-on ({{0..1} × {0..1}}) h
  assumes h ' ({{0..1} × {0..1}}) ⊆ s
  assumes ⋀x. x ∈ {0..1} ⇒ h (0, x) = p x
  assumes ⋀x. x ∈ {0..1} ⇒ h (1, x) = q x
  assumes ⋀x. x ∈ {0..1} ⇒ pathstart (h ⪫ Pair x) = pathstart p
  assumes ⋀x. x ∈ {0..1} ⇒ pathfinish (h ⪫ Pair x) = pathfinish p
  shows homotopic-paths s p q
  using assms unfolding homotopic-paths by (intro exI[of - h]) auto

lemma sum-upto-ln-conv-sum-upto-mangoldt:
  sum-upto (λn. ln (real n)) x = sum-upto (λn. mangoldt n * nat ⌊x / real n⌋) x
  proof –
    have sum-upto (λn. ln (real n)) x =
    sum-upto (λn. ∑ d | d dvd n. mangoldt d) x
    by (intro sum-upto-cong) (simp add: mangoldt-sum)
    also have . . . = sum-upto (λk. sum-upto (λd. mangoldt k) (x / real k)) x
    by (rule sum-upto-sum-divisors)
    also have . . . = sum-upto (λn. mangoldt n * nat ⌊x / real n⌋) x
    unfolding sum-upto-altdef by (simp add: mult-ac)
    finally show ?thesis .
qed

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

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

lemma \text{fds-converges-scaleR} [intro]:
\text{ assumes } \text{fds-converges } f \text{ s}
\text{ shows } \text{fds-converges } (c \ast_R f) \text{ s}
proof -
from \text{assms} have \text{summable } (\lambda n. c \ast_R (\text{fds-nth } f n / \text{nat-power } n s))
by (intro \text{summable-scaleR-right}) (auto simp: \text{fds-converges-def})
also have \((\lambda n. c \ast_R (\text{fds-nth } f n / \text{nat-power } n s)) = (\lambda n. (c \ast_R \text{fds-nth } f n / \text{nat-power } n s))\)
by (simp add: scaleR-conv-of-real)
finally show \?thesis by (simp add: \text{fds-converges-def})
qed

lemma \text{fds-abs-converges-scaleR} [intro]:
\text{ assumes } \text{fds-abs-converges } f \text{ s}
\text{ shows } \text{fds-abs-converges } (c \ast_R f) \text{ s}
proof -
from \text{assms} have \text{summable } (\lambda n. \text{abs } c \ast \text{norm } (\text{fds-nth } f n / \text{nat-power } n s))
by (intro \text{summable-mult}) (auto simp: \text{fds-abs-converges-def})
also have \((\lambda n. \text{abs } c \ast \text{norm } (\text{fds-nth } f n / \text{nat-power } n s)) = (\lambda n. \text{norm } ((c \ast_R \text{fds-nth } f n) / \text{nat-power } n s))\) by (simp add: norm-divide)
finally show \?thesis by (simp add: \text{fds-abs-converges-def})
qed

lemma \text{conv-abscissa-scaleR}: \text{conv-abscissa } (scaleR c \text{ f}) \leq \text{conv-abscissa } f
by (rule \text{conv-abscissa-mono}) auto
lemma abs-conv-abscissa-scaleR: abs-conv-abscissa (scaleR c f) ≤ abs-conv-abscissa f
  by (rule abs-conv-abscissa-mono) auto

lemma fds-conv-weak-fds.abs-converges-right-right [intro]:
  fds-conv-weak-fds.abs-converges f s ==> fds-conv-weak-fds.abs-converges (fds-const c * f) s
  by (auto simp: fds-conv-weak-fds.abs-converges-def norm-mult norm-divide dest: summable-mult[of - norm c])

lemma conv-abscissa-mono: conv-abscissa f b ≤ conv-abscissa f
  by (intro 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: \( n \rightarrow \| f(n) \|_{S} \leq C \)
    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 (\( n \rightarrow \| f(n) \|_{S} \leq fds-nth (C *R fds-zeta) n \))
        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 $Bseq (\lambda n. fds\text{-}n\text{th} f n * nat\text{-}power n s0) s \cdot 1 > 1 - s0 \cdot 1$
shows $fds\text{-}abs\text{-}converges f s$

proof
have $fds\text{-}n\text{th} (fds\text{-}shift s0 f) = (\lambda n. fds\text{-}n\text{th} f n * nat\text{-}power n s0)$
  by (auto simp: fun_eq_iff)
with assms have $Bseq (fds\text{-}n\text{th} (fds\text{-}shift s0 f))$ by simp
with assms(2) have $fds\text{-}abs\text{-}converges (fds\text{-}shift s0 f) (s + s0)$
  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 $c :: \text{ereal}$
assumes $Bseq (\lambda n. fds\text{-}n\text{th} f n)$
shows $abs\text{-}conv\text{-}abscissa f \leq c$
proof (rule abs-conv-abscissa-leI-weak)
fix $x$ assume $c < \text{ereal } x$
have $\text{ereal } (1 - s \cdot 1) \leq c$ by fact
also have $\ldots < \text{ereal } x$ by fact
finally have $1 - s \cdot 1 < \text{ereal } x$ by simp
thus $fds\text{-}abs\text{-}converges f (of\text{-}real } x)$
  by (intro bounded-coeffs-imp-fds-abs-converges[where $s = 0$]) auto
qed

lemma bounded-coeffs-imp-abs-conv-abscissa-le-1:
fixes $s :: 'a :: dirichlet-series$ and $f :: 'a fds$
assumes $\text{Bseq } (\lambda n. fds\text{-}n\text{th} f n)$
shows $\text{abs\text{-}conv\text{-}abscissa } f \leq 1$
proof
have $[simp]: fds\text{-}n\text{th} f n * nat\text{-}power n 0 = fds\text{-}n\text{th} f n$ 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 $a \ b \ c :: \text{real}$
assumes $\text{abs: } a + b > 0 \text{ and } c < -1$
shows $\text{set\text{-}integrable\text{-}powr\text{-}at\text{-}top}: (\lambda x. (b + x) \text{ powr} c) \text{ absolutely\text{-}integrable\text{-}on } \{a <..\}$
  and $\text{set\text{-}lebesgue\text{-}integral\text{-}powr\text{-}at\text{-}top}$:
  $((b + x) \text{ powr} c) \partial \text{borel}) = -((b + a) \text{ powr} (c + 1) / (c + 1))$
  and $\text{powr\text{-}has\text{-}integral\text{-}at\text{-}top}$:
  $((\lambda x. (b + x) \text{ powr} c) \text{ has\text{-}integral } -((b + a) \text{ powr} (c + 1) / (c + 1)))$
  \{a <..\}
proof –
  let \( ?f = \lambda x. (b + x) \text{ powr } (c + 1) \) and \( ?F = \lambda x. (b + x) \text{ powr } (c + 1) / (c + 1) \)
  have limits: ((\( ?F \circ \text{ real-of-ereal} \)) \( \longrightarrow ?F a \)) (at-right (ereal a))
  \((\( ?F \circ \text{ real-of-ereal} \)) \( \longrightarrow 0 \)) (at-left (\text{ereal} a))
  using c ab unfolding ereal-tendsto-simps1 by (real-asym simp: field-simps)+
  have 1: set-integrable lborel (ieinterval a ∞) \( ?f \) using ab c limits
    by (intro interval-integral-FTC-nonneg) (auto intro!: derivative-eq-intros)
  thus 2: \( ?f \) absolutely-integrable-on \( \{a<..\} \)
    by (auto simp add: set-integrable-def integrable-completion)
  have \( \text{ LBINT x=ereal a..∞. (b+x) powr c = 0 - ?F a} \) using ab c limits
    by (intro interval-integral-FTC-nonneg) (auto intro!: derivative-eq-intros)
  thus 3: \( \int x\in\{a<..\}. ((b+x) powr c) \, \text{d}borel = -(\( b + a \) powr (c + 1) / (c + 1)) \)
    by (simp add: interval-integral-to-infinity-eq)
  show (\( ?f \) has-integral \( -(\( b + a \) powr (c + 1) / (c + 1)) \) \( \{a<..\} \))
    using set-borel-integral-eq-integral[OF 1] 3 by (simp add: has-integral-iff)
  qed

lemma fds-converges-altdef2:
  fds-converges f s ←→ convergent (λN. eval-fds (fds-truncate N f) s)
unfolding fds-converges-def summable-iff-convergent' eval-fds-truncate
by (auto simp: not-le intro!: convergent-cong always-eventually sum mono-neutral-right)

lemma tendsto-eval-fds-truncate:
  assumes fds-converges f s
  shows (λN. eval-fds (fds-truncate N f) s) \( \longrightarrow \) eval-fds f s
proof –
  have (λN. eval-fds (fds-truncate N f) s) \( \longrightarrow \) eval-fds f s ←→
    (λN. \( \sum i\le N. \) fds-nth f i / nat-power i s) \( \longrightarrow \) 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 ?thesis .
  qed

lemma linepath-translate-left: linepath (c + a) (c + a) = (λx. c + a) \circ linepath a b
  by auto

lemma linepath-translate-right: linepath (a + c) (b + c) = (λx. x + c) \circ linepath a b
  by (auto simp: fun-eq_iff linepath-def algebra-simps)

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

**proof**

- **have** deriv: vector-derivative \(((\lambda x. x - \text{Im } a \ast i) \circ \text{linepath } a \ b) \ (\text{at } y) = b - a\)

**for** \(y\)

- **using** linepath-translate-right[of a \(-\text{Im } a \ast i \ b\), symmetric] **by** simp

**have** \((f \text{ has-contour-integral } I) \ (\text{linepath } a \ b) \iff ((\lambda x. f (x + \text{Im } a \ast i)) \text{ has-contour-integral } I) \ (\text{linepath } (a - \text{Im } a \ast i) (b - \text{Im } a \ast i))\)

**using** linepath-translate-right[of a \(-\text{Im } a \ast i \ b\)] **deriv** by (simp add: has-contour-integral)

**also have** \(\ldots \iff ((\lambda x. f (x + \text{Im } a \ast i)) \text{ has-integral } I) \ \{\text{Re } a..\text{Re } b\}\ **using**\)

**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** \(a \ b :: \text{complex and } f :: \text{complex} \Rightarrow \text{complex}\)

**assumes** \(\text{Im } a = \text{Im } b \ \text{Re } a < \text{Re } b\)

**shows** \((f \text{ contour-integrable-on linepath } a \ b) \iff (\lambda x. f (x + \text{Im } a \ast i)) \text{ integrable-on } \{\text{Re } a..\text{Re } b\}\)

**using** contour-integrable-on-def has-contour-integral-linepath-same-Im-iff[OF assms]

**by** blast

**lemma** contour-integral-linepath-same-Im:

- **fixes** \(a \ b :: \text{complex and } f :: \text{complex} \Rightarrow \text{complex}\)

**assumes** \(\text{Im } a = \text{Im } b \ \text{Re } a < \text{Re } b\)

**shows** \(\text{contour-integral } (\text{linepath } a \ b) f = \text{integral } \{\text{Re } a..\text{Re } b\} (\lambda x. f (x + \text{Im } a \ast i))\)

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

**case** True

- **thus** \(?\text{thesis} \text{ using } \text{has-contour-integral-linepath-same-Im-iff}[\text{OF assms, of } f] \)

**using** has-contour-integral-has-contour-integral-unique **by** blast

**next**

**case** False

- **thus** \(?\text{thesis} \text{ using } \text{contour-integrable-linepath-same-Im-iff}[\text{OF assms, of } f] \)

**by** (simp add: not-integrable-contour-integral not-integrable-integral)

**qed**

**lemmas** [simp del] = div-mult-self3 div-mult-self4 div-mult-self2 div-mult-self1

**interpretation** cis: periodic-fun-simple cis 2 * pi

- **by** standard (simp-all add: complex-eq-iff)

**lemma** analytic-onE-box:

- **assumes** \(f \text{ analytic-on } A \ s \in A\)

**obtains** \(a \ b\) **where** \(\text{Re } a < \text{Re } b \ \text{Im } a < \text{Im } b \ s \in \text{box } a \ b \ f \text{ analytic-on box } a \ b\)

**proof**

- **from** assms **obtain** \(r\) **where** \(r > 0\ f \text{ holomorphic-on ball } s \ r\)
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 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 Re-image-cbox:
assumes Re a ≤ Re b Im a ≤ Im b
shows Re ' cbox a b = {Re a..Re b}
using inner-image-cbox[of 1::complex a b] assms by (auto simp; Basis-complex-def)

lemma Im-image-cbox:
assumes Re a ≤ Re b Im a ≤ 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 ∈ 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 (\(\forall x. x \in A \implies f x \neq 1\)) a > 0
  shows \((\lambda x. hurwitz-zeta a (f x))\) 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 (\(\forall x. x \in A \implies f x \neq 1\))
  shows \((\lambda x. zeta (f x))\) 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: \((\lambda s. deriv zeta s / zeta s)\) analytic-on \(\{s. Re s \geq 1\}\) \(-\{1\}\)
  using zeta-Re-ge-1-nonzero by (auto intro!: analytic-intros)

lemma mult-real-sqrt: \(x \geq 0 \implies x \cdot sqrt y = sqrt (x ^ 2 \cdot y)\)
  by (simp add: real-sqrt-mult)

lemma arcsin-pos: \(x \in \{0<..1\} \implies arcsin x > 0\)
  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 \(\in s\) holomorphic-on s
  shows residue (\(\lambda w. f w / w\)) 0 = f 0
  using residue-simple[of s 0 f] assms by simp

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

lemma conv-abscissa-cong:
  assumes eventually (\(\lambda n. fds-nth f n = fds-nth g n\)) at-top
  shows conv-abscissa f = conv-abscissa g
proof
have fds-converges f = 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 (λn. fds-nth f n = fds-nth g n) at-top
  shows abs-conv-abscissa f = abs-conv-abscissa g
proof
have fds-abs-converges f = 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
  fds-remainder m = fds-subseries (λn. n > m)

lemma fds-nth-remainder: fds-nth (fds-remainder m f) = (λn. if n > m then fds-nth f n else 0)
  by (simp add: fds-remainder-def fds-subseries-def fds-nth-fds)

lemma fds-converges-remainder-iff [simp]:
  fds-converges (fds-remainder m f) s ⇔ 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]:
  fds-abs-converges (fds-remainder m f) s ⇔ 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]:
  fds-converges f s ⇒ fds-converges (fds-remainder m f) s
and fds-abs-converges-remainder [intro]:
  fds-abs-converges f s ⇒ fds-abs-converges (fds-remainder m f) s
  by simp-all

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

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-nth-remainder)
lemma eval-fds-remainder:
  \[ \text{eval-fds (fds-remainder m f) s = } \sum n. \text{fds-nth (f + Suc m) } (n + \text{Suc m}) \ / \text{nat-power } (n + \text{Suc m}) \ s \] 
  \( (\text{s s = suminf (\lambda \cdot \text{fds-nth (f + Suc m))})} \)

proof (cases fds-converges f s)
  case False
  hence ~fds-converges (fds-remainder m f) s by simp
  hence \((\lambda x. \text{fds-nth (f + Suc m) } (n + \text{nat-power n s) sums x}) = (\lambda \cdot \text{False}) \)
  by (auto simp: fds-converges-def summable-def)
  hence eval-fds (fds-remainder m f) s = (THE - False) by simp
  moreover from False have ~summable (\lambda n. \text{fds-nth (f + Suc m)}) unfolding fds-converges-def
  by (subst suminf-minus-initial-segment) (auto simp: summable-def)
  ultimately show \?thesis by simp
  next
  case True
  hence *: fds-converges (fds-remainder m f) s by simp
  have eval-fds (fds-remainder m f) s = (\sum n. \text{fds-nth (f + Suc m) } (n + Suc m) \ s) unfolding eval-fds-def ..
  also have \ldots = (\sum n. \text{fds-nth (f + Suc m) } (n + Suc m) \ s) unfolding eval-fds-def ..
  using * unfolding fds-converges-def
  by (auto simp: fds-nth-remainder)
  also have (\lambda n. \text{fds-nth (f + Suc m)}) = (\lambda n. \text{fds-nth f } (n + Suc m))
  by (intro ext) (auto simp: fds-nth-remainder)
  finally show ?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 \(x. x \in A \implies Re (g x) > conv-abscissa f\)
  shows (\lambda x. \text{eval-fds (f g x)}) holomorphic-on A
  using holomorphic-on-compose-gen[OF assms(1) holomorphic-fds-eval[OF order.refl, of f]] assms(2)
  by (auto simp: o-def)

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

using analytic-on-compose-gen[OF assms(1) analytic-fds-eval[OF order_refl, of \(f\)]

by (auto simp: o-def)

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 \(b1 = a1 + 2 * \text{of-int } k * \pi\) and \(b2 = a2 + 2 * \text{of-int } k * \pi\)
shows homotopic-loops \(A\) (part-circlepath \(c\) \(r\) \(a1\) \(b1\)) (part-circlepath \(c\) \(r\) \(a2\) \(b2\))
proof
  define \(h\) where \(h = (\lambda (x,y). \text{part-circlepath} \ (c \ x) \ (r \ x) \ (a \ x) \ (b \ x) \ (f \ x))\)
  show \(?thesis\)
  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 \(h \cdot \{0..1\} \times \{0..1\}\) \(\subseteq\) sphere \(c\) \(r\)
      by (auto simp: h-def part-circlepath-def dist-norm norm-mult)
    also have \(\ldots \subseteq\) \(A\) by fact
    finally show \(h \cdot \{0..1\} \times \{0..1\}\) \(\subseteq\) \(A\) .
  next
  fix \(x\) :: real
  assume \(x\) : \(x \in \{0..1\}\)
  show \(h \ (0, x) = \text{part-circlepath} \ c \ r \ (a1 \ b1 \ x)\) and \(h \ (1, x) = \text{part-circlepath} \ c \ r \ (a2 \ b2 \ x)\)
    by (simp-all add: h-def linepath-def)
  have \(\text{cis} \ (\pi * (\text{real-of-int} \ k * 2)) = 1\)
    using cis.plus-of-int[of 0 k] by (simp add: algebra-simps)
  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)
  qed
qed

lemma part-circlepath-cone-subpath:
part-circlepath \(c\) \(r\) \(a\) \(b\) = subpath \((a \ / \ (2 * \pi)) \ (b \ / \ (2 * \pi))\) (circlepath \(c\) \(r\))
by (simp add: part-circlepath-def circlepath-def subpath-def linepath-def algebra-simps)
lemma homotopic-paths-part-circlepath:
  assumes a ≤ b b ≤ c
  assumes path-image (part-circlepath C r a c) ⊆ A r ≥ 0
  shows homotopic-paths A (part-circlepath C r a c)
  (part-circlepath C r a b +++ part-circlepath C r b c)
  (is homotopic-paths · ?g (??h1 +++ ??h2))
proof (cases a = c)
case False
  with assms have a < c by simp
  define slope where slope = (b − a) / (c − a)
  from assms and ⟨a < c⟩ have slope: slope ∈ {0..1}
  by (auto simp: field-simps slope-def)
  define f :: real ⇒ real where f = linepath 0 slope +++ linepath slope 1
  show ?thesis
  proof (rule homotopic-paths-reparametrize)
    fix t :: real assume t: t ∈ {0..1}
    show (??h1 +++ ??h2) t = ?g (f t)
    proof (cases t ≤ 1 / 2)
    case True
      hence ?g (f t) = C + r * cis ((1 − f t) * a + f t * c)
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      also from True ⟨a < c⟩ have (1 − f t) * a + f t * c = (1 − 2 * t) * a + 2 * t * b
      unfolding f-def slope-def linepath-def joinpaths-def
      by (simp add: divide-simps del: div-mult-self3 div-mult-self4 div-mult-self2
      div-mult-self1)
      (simp add: algebra-simps)?
      also from True have C + r * cis ... = (?h1 +++ ?h2) t
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      finally show ?thesis ..
    next
    case False
      hence ?g (f t) = C + r * cis ((1 − f t) * a + f t * c)
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      also from False ⟨a < c⟩ have (1 − f t) * a + f t * c = (2 − 2 * t) * b +
      (2 * t − 1) * c
      unfolding f-def slope-def linepath-def joinpaths-def
      by (simp add: divide-simps del: div-mult-self3 div-mult-self4 div-mult-self2
      div-mult-self1)
      (simp add: algebra-simps)?
      also from False have C + r * cis ... = (?h1 +++ ?h2) t
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      finally show ?thesis ..
  qed
next
from slope have \( path-image f \subseteq \{0..1\} \)
by (auto simp: f-def path-image-join closed-segment-eq-real-ivl)
thus \( f \in \{0..1\} \rightarrow \{0..1\} \) by (force simp add: path-image-def)

next
have \( path f \) unfolding f-def by auto
thus continuous-on \( \{0..1\} \) \( f \) by (simp add: path-def)

next
case [simp]: True
with assms have [simp]: \( b = c \) by auto
have \( \text{part-circlepath } C r c c + + + \text{ part-circlepath } C r c c = \text{ part-circlepath } C r c c \)
by (simp add: path-def)
thus \(?thesis\) using assms by simp

lemma \( \text{path-image-part-circlepath-subset}: \)
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 assms by (subst (1 2) path-image-part-circlepath) auto

lemma \( \text{part-circlepath-mirror}: \)
assumes \( a' = a + pi + 2 * pi * \text{of-int } k b' = b + pi + 2 * pi * \text{of-int } k c' = -c \)
shows \( -\text{part-circlepath } c r a b = \text{part-circlepath } c' r a' b' \)
proof
fix \( x :: \text{real} \)
have \( \text{part-circlepath } c' r a' b' x = c' + r * \text{cis } (\text{linepath } a b x + pi + k * (2 * pi)) \)
by (simp add: part-circlepath-def exp-eq-polar assms linepath-translate-right mult-ac)
also have \( \text{cis } (\text{linepath } a b x + pi + k * (2 * pi)) = \text{cis } (\text{linepath } a b x + pi) \)
by (rule cis.plus-of-int)
also have \( \ldots = -\text{cis } (\text{linepath } a b x) \)
by (simp add: minus-cis)
also have \( \ldots = -\text{part-circlepath } c r a b x \)
by (simp add: part-circlepath-def assms 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 \( \text{path-mirror } [\text{intro}]: \text{path } (g :: - \Rightarrow 'b::topological-group-add) \Longrightarrow \text{path } (-g) \)
by (auto simp: path-def intrl: continuous-intros)

lemma \( \text{path-mirror-iff } [\text{simp}]: \text{path } (-g :: - \Rightarrow 'b::topological-group-add) \iff \text{path } g \)
using \( \text{path-mirror[of } g \text{] path-mirror[of } -g \text{]} \) by (auto simp: fun-Compl-def)
lemma valid-path-mirror [intro]: valid-path g \implies valid-path (−g)
by (auto simp: valid-path-def fun-Compl-def piecewise-C1-differentiable-neg)

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 cos-le-zero:
assumes x ∈ {pi/2..3*pi/2}
shows cos x ≤ 0
proof
have cos x = −cos (x − pi) by (simp add: cos-diff)
moreover from assms have cos (x − pi) ≥ 0
by (intro cos-ge-zero) auto
ultimately show ?thesis by simp
qed

lemma cos-le-zero': x ∈ {−3*pi/2..−pi/2} \implies cos x ≤ 0
using cos-le-zero[of −x] by simp

lemma winding-number-join-pos-combined':
[valid-path γ1 \land z \notin path-image γ1 \land 0 < Re (winding-number γ1 z);
valid-path γ2 \land z \notin path-image γ2 \land 0 < Re (winding-number γ2 z);
pathfinish γ1 = pathstart γ2]
\implies valid-path(γ1 +++ γ2) \land z \notin path-image(γ1 +++ γ2) \land 0 < Re(winding-number(γ1 +++ γ2) 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
shows (⋃n∈{a..<b}. {real n..real (n + 1)}) = {real a..real b}
proof (intro equalityI subsetI)
fix x assume x: x ∈ {real a..real b}
thus x ∈ (⋃n∈{a..<b}. {real n..real (n + 1)})
proof (cases x = real b)
  case True
  with assms show ?thesis by (auto intro!: bexI[of - b - 1])
next
case False
with x have x: x ≥ real a \land x < real b by simp-all
hence x ≥ real (nat [x]) \land x ≤ real (Suc (nat [x])) by linarith+
moreover from x have nat [x] ≥ a \land nat [x] < b by linarith+
ultimately show ?thesis by force
lemma nat-sum-has-integral-floor:
fixes f :: nat ⇒ 'a :: banach
assumes mn: m < n
shows (λx. f (nat ⌈x⌉)) \text{has integral} f \{m..<n\} \real m..real n
proof –
define D where D = (λi. {real i..real (Suc i)}) \{m..<n\}
have D: D \text{division-of} \{m..n\}
  using Union-atLeastAtMost-real-of-nat[OF mn] by (simp add: division-of-def D-def)
have ((λx. f (nat ⌈x⌉)) \text{has integral} (∑x∈D. f (nat [Inf X]))) \{real m..real n\}
proof (rule has-integral-combine-division)
fix X assume X: X ∈ D
have nat ⌈x⌉ = nat [Inf X] if x ∈ X – {Sup X} for x
  using that X by (auto simp: D-def nat-eq-iff floor-eq-iff)
hence ((λx. f (nat ⌈x⌉)) \text{has integral} f (nat [Inf X])) X \iff
  ((λx. f (nat [Inf X])) \text{has integral} f (nat [Inf X])) X \text{using} X
by (intro has-integral-spike-eq[of {Sup X}]) auto
also from X have \ldots using has-integral-const-real[of f (nat [Inf X]) Inf X Sup X]
  by (auto simp: D-def)
finally show (λx. f (nat ⌈x⌉)) \text{has integral} f (nat [Inf X]) X .
qed

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

fact+
also have \((\sum X \in D. f (\text{natsup} X)) = (\sum k \in \{m<..n\}. f (\text{Suc} k))\)
unfolding D-def by (subst sum.reindex) (auto simp: inj-on-def nat-add-distrib)
also have \(\ldots = (\sum k \in \{m<..n\}. f k)\)
by (intro sum.reindex-bij-witness[of - \lambda x. x - 1 Suc]) auto
finally show \(?thesis\).
qed

lemma zeta-partial-sum-le:
fixes \(x \::\) real and \(m \::\) nat
assumes \(x \in \{0<..1\}\)
shows \((\sum n = 1..m. \text{natsup} k \text{ powr} (x - 1)) \leq \text{natsup} m \text{ powr} x / x\)
proof
consider \(m = 0 \mid m = 1 \mid m > 1\) by force
thus \(?thesis\)
proof cases
  assume \(m: m > 1\)
  hence \(\{1..m\} = \text{insert} \ 1 \ \{1<..m\}\) by auto
  also have \((\sum k \in \{1<..m\}. \text{natsup} k \text{ powr} (x - 1)) = 1 + (\sum k \in \{1<..m\}. \text{natsup} k \text{ powr} (x - 1))\)
  by simp
  also have \((\sum k \in \{1<..m\}. \text{natsup} k \text{ powr} (x - 1)) \leq \text{natsup} m \text{ powr} x / x - 1 / x\)
  proof (rule has-integral-le)
    show \((\lambda t. (\text{natsup} \ t) \text{ powr} (x - 1)) \text{ has-integral} (\sum n \in \{1<..m\}. n \text{ powr} (x - 1))) \ \{\text{real} 1..m\}\)
    using \(m\) by (intro nat-sum-has-integral-ceiling) auto
  next
    have \((\lambda t. t \text{ powr} (x - 1)) \text{ has-integral} (\text{natsup} m \text{ powr} x / x - \text{natsup} 1 \text{ powr} x / x)\)
    \{\text{real} 1..\text{natsup} m\}\)
    by (intro fundamental-theorem-of-calculus)
    (insert \(x\ m\), auto simp flip: has-real-derivative-iff-has-vector-derivative
        intro!: derivative-eq-intros)
    thus \((\lambda t. t \text{ powr} (x - 1)) \text{ has-integral} (\text{natsup} m \text{ powr} x / x - 1 / x)\) \ {\text{real} 1..\text{natsup} m}\)
    by simp
  qed
  also have \(1 + (\text{natsup} m \text{ powr} x / x - 1 / x) \leq \text{natsup} m \text{ powr} x / x\)
  using \(x\) by (simp add: field-simps)
  finally show \(?thesis\) by simp
qed (use assms in auto)
qed

lemma zeta-partial-sum-le':
fixes \(x \::\) real and \(m \::\) nat
assumes \(x > 0\) and \(m > 0\)
shows \((\sum n = 1..m. \text{natsup} n \text{ powr} (x - 1)) \leq \text{natsup} m \text{ powr} x * (1 / x + 1 / m)\)
proof (cases x > 1)
  case False
  with assms have \((\sum_{n=1..m} n \text{ powr } (x - 1)) \leq m \text{ powr } x / x\)
  by (intro zeta-partial-sum-le) auto
  also have \(\ldots \leq m \text{ powr } x * (1 / x + 1 / m)\)
  using assms by (simp add: field-simps)
  finally show \(?thesis\).
next
  case True
  have \((\sum_{n\in\{1..m\}} n \text{ powr } (x - 1)) = (\sum_{n\in\text{insert } m \{0..m\}. n \text{ powr } (x - 1)})\)
  by (intro sum.mono-neutral-left) auto
  also have \(\ldots = m \text{ powr } (x - 1) + (\sum_{n\in\{0..m\}. n \text{ powr } (x - 1)})\) by simp
  also have \((\sum_{n\in\{0..m\}. n \text{ powr } (x - 1)}) \leq \text{ real } m \text{ powr } x / x\)
  proof
    show \(((\lambda t. (\text{nat } \lfloor t \rfloor) \text{ powr } (x - 1)) \text{ has-integral } (\sum_{n\in\{0..m\}. n \text{ powr } (x - 1)})\) \{\text{real } 0..m\}\)
    using m by (intro intro-nat-sum-has-integral-floor) auto
  next
    show \(((\lambda t. \text{ powr } (x - 1)) \text{ has-integral } (\text{ real } m \text{ powr } x / x)\) \{\text{real } 0..\text{real } m\}\)
    using has-integral-powr-from-0[of x - 1] x by auto
  next
    fix t assume t \in\{\text{real } 0..\text{real } m\}
    with \(x > 1\) 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 \(m \text{ powr } (x - 1) + m \text{ powr } x / x = m \text{ powr } x * (1 / x + 1 / m)\)
  using m x by (simp add: powr-diff field-simps)
  finally show \(?thesis\) by simp
  qed

lemma natfun-bigo-1E:
  assumes \(f :: \text{nat } \Rightarrow \cdot \in O(\lambda x. 1)\)
  obtains \(C\) where \(\text{C \geq lb \land n. norm } (f n) \leq C\)
proof
  from assms obtain \(C N\) where \(\forall n \geq N. \text{ norm } (f n) \leq C\)
  by (auto elim!: landau-o.bigE simp: eventually-at-top-linorder)
  hence \(\text{norm } (f n) \leq \text{Max } (\{C, lb\} \cup (\text{norm } f \cdot \{..<N\}))\) for n
  by (cases n \geq N) (subst Max-ge-iff; force simp: image-iff)+
  moreover have \(\text{Max } (\{C, lb\} \cup (\text{norm } f \cdot \{..<N\})) \geq lb\)
  by (intro Max.coboundedI) auto
  ultimately have \(\text{Max } (\{C, lb\} \cup (\text{norm } f \cdot \{..<N\})) \geq lb\)
  by (intro Max.coboundedI) auto
  ultimately show \(?thesis\) using that by blast
  qed

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

19
next
assume $f \in O(\lambda x. 1)$
from natfun-bigo-1E[OF this, where lb = 1] obtain $C$ where $C \geq 1$ \wedge \text{n. 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 $f :: \text{real} \Rightarrow \text{ennreal}$
assumes decreasing: $\forall x \ y. 0 \leq x \Longrightarrow x \leq y \Longrightarrow f y \leq f x$
shows $(\sum n. f (\text{real} (\text{Suc} n))) \leq \text{set-nn-integral lborel} \{0..\} f$
proof
have $(\sum n. f (\text{Suc} n))) = (\sum n. f (\text{real} <..\text{real} (\text{Suc} n)). (\text{Suc} n) \partial \text{lborel})$
by (subst nn-integral-cmult-indicator) auto
also have nat $[x] = \text{Suc} n$ if $x \in \{\text{real} <..\text{real} (\text{Suc} n)\}$ for $x \ n$
using that by (auto simp: nat-eq-iff ceiling-eq-iff)
hence $(\sum n. f + x \in \{\text{real} <..\text{real} (\text{Suc} n)\}). (\text{Suc} n) \partial \text{lborel}) = (\sum n. f + x \in \{\text{real} <..\text{real} (\text{Suc} n)\}). (\text{real} (\text{nat} [x])) \partial \text{lborel})$
by (intro suminf-cong nn-integral-cong) (auto simp: indicator-def)
also have \dots $= (f + x \in (\bigcup i \in \{\text{real} i<..\text{real} (\text{Suc} i)\}). (\text{real} [x:real]) \partial \text{lborel})$
by (subti nn-integral-disjoint-family)
(auto simp: disjoint-family-on-def)
also have \dots $\leq (f + x \in \{0..\}. (f x) \partial \text{lborel})$
by (intro nn-integral-mono) (auto simp: indicator-def intro!: decreasing)
finally show \text{thesis} .
qed

lemma abs-summable-on-aminus-iff:
$(\lambda x. -f x)$ abs-summable-on $A \longleftrightarrow f$ abs-summable-on $A$
by (simp add: abs-summable-on-def)

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 $(\lambda x. c \ast f x)$ abs-summable-on $A \longleftrightarrow f$ abs-summable-on $A$
by (simp add: abs-summable-on-alfdef assms)

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 $(\lambda x. f x \ast c)$ abs-summable-on $A \longleftrightarrow f$ abs-summable-on $A$
by (simp add: abs-summable-on-alfdef assms)

lemma decreasing-sum-le-integral:
fixes $f :: \text{real} \Rightarrow \text{real}$
assumes nonneg: $\forall x. x \geq 0 \Longrightarrow f x \geq 0$
assumes decreasing: $\forall x \ y. 0 \leq x \Longrightarrow x \leq y \Longrightarrow f y \leq f x$

20
assumes \( \text{integral: } \{ f \text{ has-integral } I \} \{ \theta .. \} \)

shows \( \text{summable } (\lambda i. f (\text{real } (\text{Suc } i))) \) and \( \text{suminf } (\lambda i. f (\text{real } (\text{Suc } i))) \leq I \)

proof –

have \([\text{simp}]: I \geq 0\)
  by \((\text{intro has-integral-nonneg} [\text{OF integral} \text{ nonneg}] \text{ auto})\)

have \((\sum n. \text{ennreal } (f (\text{Suc } n))) =
  (\sum n. \int^+ x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}. \text{ennreal } (f (\text{Suc } n)) \; \partial \text{borel})\)
  by \((\text{subst nn-integral-cmult-indicator}) \text{ auto})\)

also have \(\text{nat } [x] = \text{Suc } n \text{ if } x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\} \text{ for } x n\)

using that by \((\text{auto simp: nat-eq-iff ceiling-eq-iff})\)

hence \((\sum n. \int^+ x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}. \text{ennreal } (f (\text{Suc } n)) \; \partial \text{borel}) =
  (\sum n. \int^+ x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}. \text{ennreal } (f (\text{real } (\text{nat } [x]))) \; \partial \text{borel})\)
  by \((\text{intro suminf-cong nn-integral-cong}) \text{ (auto simp: indicator-def})\)

also have \(\ldots = (\int^+ x \in (\bigcup i. \{\text{real } i <.. \text{real } (\text{Suc } i)\}). \text{ennreal } (f (\text{nat } [x::\text{real}])) \partial \text{borel})\)

by \((\text{subst nn-integral-disjoint-family})\)

\((\text{auto simp: disjoint-family-on-def intro!: measurable-completion})\)

also have \(\ldots \leq (\int^+ x \in (\ldots). \text{ennreal } (f x) \; \partial \text{borel})\)

by \((\text{intro nn-integral-mono}) \text{ (auto simp: indicator-def nonneg intro!: decreasing})\)

also have \(\ldots = (\int^+ x. \text{ennreal } (\text{indicat-real } (\ldots) \; x * f x) \; \partial \text{borel})\)

by \((\text{intro nn-integral-cong}) \text{ (auto simp: indicator-def})\)

also have \(\ldots = \text{ennreal } I\)

using \(\text{nn-integral-has-integral-lebesgue} [\text{OF nonneg integral}] \text{ by (auto simp: nonneg})\)

finally have \(*: (\sum n. \text{ennreal } (f (\text{Suc } n))) \leq \text{ennreal } I.\)

from * show \(\text{summable: summable } (\lambda i. f (\text{real } (\text{Suc } i)))\)

by \((\text{intro summable-suminf-not-top}) \text{ (auto simp: top-unique intro: nonneg})\)

note *

also from \(\text{summable have } (\sum n. \text{ennreal } (f (\text{Suc } n))) = \text{ennreal } (\sum n. f (\text{Suc } n))\)

by \((\text{subst suminf-ennreal2}) \text{ (auto simp: o-def nonneg})\)

finally show \((\sum n. f (\text{real } (\text{Suc } n))) \leq I \text{ by (subst (asm) ennreal-le-iff}) \text{ auto})\)

qed

lemma decreasing-sum-le-integral:\)

fixes \(f : \text{real } \Rightarrow \text{real}\)

assumes \(\forall x. x \geq 0 \Rightarrow f x \geq 0\)

assumes \(\forall x y. 0 \leq x \Rightarrow x \leq y \Rightarrow f y \leq f x\)

assumes \((f \text{ has-integral } I) \{0 .. \}\)

shows \(\text{summable } (\lambda i. f (\text{real } i)) \) and \(\text{suminf } (\lambda i. f (\text{real } i)) \leq f 0 + I\)

proof –

have \(\text{summable } (\langle \lambda i. f (\text{real } i) \rangle)\)

using \(\text{decreasing-sum-le-integral} [\text{OF assms}] \text{ by (simp add: o-def})\)

thus *: \(\text{summable } (\lambda i. f (\text{real } i)) \text{ by (subst (asm) summable-Suc-iff})\)

have \((\sum n. f (\text{real } (\text{Suc } n))) \leq I \text{ by (intro decreasing-sum-le-integral assms})\)

thus \(\text{suminf } (\lambda i. f (\text{real } i)) \leq f 0 + I\)

using * by \((\text{subst (asm) suminf-split-head}) \text{ auto})\)

qed
lemma of-nat-powr-neq-1-complex [simp]:
  assumes n > 1 Re s ≠ 0
  shows of-nat n powr s ≠ (1::complex)
proof –
  have norm (of-nat n powr s) = real n powr Re s
    by (simp add: norm-powr-real-powr)
  also have ... ≠ 1
    using assms by (auto simp: powr-def)
finally show ?thesis by auto
qed

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

lemma fds-nth-logderiv-completely-multiplicative:
  fixes f :: 'a :: {real-normed-field} fds
  assumes completely-multiplicative-function (fds-nth f) fds-nth f 1 ≠ 0
  shows fds-nth (fds-deriv f / f) n = - fds-nth f n * 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 :: dirichlet-series and l :: 'a and f :: 'a fds
  defines h ≡ fds-deriv f / f
  assumes completely-multiplicative-function (fds-nth f) and [simp]: fds-nth f 1 ≠ 0
  assumes s · 1 > abs-conv-abscissa f
  shows (λp. of-real (ln (real p)) * (1 / (1 - fds-nth f p / nat-power p s) - 1))
    abs-summable-on {p. prime p} (is ?th1)
    and eval-fds h s = - (∑ a p | prime p. of-real (ln (real p)) *
        (1 / (1 - fds-nth f p / nat-power p s) - 1)) (is ?th2)
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-conv-abscissa-completely-multiplicative-log-deriv[of assms(2)] assms
    by (intro fds-abs-converges) auto
  hence h ≡ (λn. fds-nth h n / nat-power n s) abs-summable-on UNIV
    by (auto simp: h-def fds-abs-converges-altdef)

note *
also have \((\lambda n. \text{fds nth } h n \div 	ext{nat-power } n s) \text{ abs-summable-on } \text{UNIV} \iff (\lambda x. -(\text{fds nth } f x * \text{mangoldt } x \div \text{nat-power } x s) \text{ abs-summable-on } \text{Collect } \text{primepow})\)

unfolding \(h\)-def using \(\text{fds nth-logderiv-completely-multiplicative}[\text{OF } \text{assms}(2)]\)
by \((\text{intro } \text{abs-summable-on-cong-neutral})\) \((\text{auto simp: } \text{fds nth-fds mangoldt-def})\)
finally have \(\text{sum1: } (\lambda x. -(\text{fds nth } f x * \text{mangoldt } x \div \text{nat-power } x s) \text{ abs-summable-on } \text{Collect } \text{primepow})\)
by \((\text{rule abs-summable-on-subset})\) \(\text{auto}\)
also have \(\beta\text{this} \iff (\lambda(p,k). -(\text{fds nth } f (p \div \text{Suc } k) * \text{mangoldt } (p \div \text{Suc } k) \div \text{nat-power } (p \div \text{Suc } k) s) \text{ abs-summable-on } (?P \times \text{UNIV})\)
using \(\text{bij-betu-primepows}\)
unfolding \(\text{case-prod-unfold}\)
also have \(\ldots \iff (\lambda(p,k). -(f \text{ds nth } f p \div \text{nat-power } p s) \div \text{Suc } k \div \text{of-real } (\ln (\text{real } p)))\)
\(\text{abs-summable-on } (?P \times \text{UNIV})\)
unfolding \(\text{case-prod-unfold}\)
by \((\text{intro } \text{abs-summable-on-cong}, \text{subst } \text{mangoldt-primepows})\)
\((\text{auto simp: } f.\text{mult } f.\text{power } \text{nat-power-power-left } \text{power-divide }\text{dest: } \text{prime-gt-1-nat})\)
finally have \(\text{sum2: } \ldots . .\)

have \(\text{sum4: } \text{summable } (\lambda n. (\text{norm } (\text{fds nth } f p \div \text{nat-power } p s)) \div \text{Suc } n) \text{ if } p: \text{prime } p \text{ for } p\)
proof -
have \(\text{summable } (\lambda n. (\ln (\text{real } p)) \div (\text{norm } (\text{fds nth } f p \div \text{nat-power } p s)) \div \text{Suc } n)\)
using \(p \text{ abs-summable-on-Sigma-project2}[\text{OF } \text{sum2, of } p]\)
unfolding \(\text{abs-summable-on-nat-iff}'\)
by \((\text{simp add: norm-power norm-mult norm-divide mult-ac del: power-Suc})\)
thus \(\beta\text{thesis add: norm-power norm-mult norm-divide mult-ac del: power-Suc}\)
by \((\text{rule summable-mult-D})\) \((\text{insert } p, \text{auto dest: prime-gt-1-nat})\)
qed

have \(\text{sums: } (\lambda n. (\text{fds nth } f p \div \text{nat-power } p s) \div \text{Suc } n) \text{ sums } (1 / (1 - \text{fds nth } f p \div \text{nat-power } p s) - 1) \text{ if } p: \text{prime } p \text{ for } p :: \text{nat}\)
proof -
from \(\text{sum4[OF } p]\)

have \(\text{norm } (\text{fds nth } f p \div \text{nat-power } p s) < 1\)
unfolding \(\text{summable-Suc-iff}\)
by \((\text{simp add: summable-geometric-iff})\)
from \(\text{geometric-sums[OF } this]\)

show \(\beta\text{thesis by } (\text{subst sums-Suc-iff})\)
auto
qed

have \(\text{eq: } (\sum a. k. -(\text{fds nth } f p \div \text{nat-power } p s) \div \text{Suc } k \div \text{of-real } (\ln (\text{real } p))) = -(\text{of-real } (\ln (\text{real } p)) \div (1 / (1 - \text{fds nth } f p \div \text{nat-power } p s) - 1))\)
if \(p: \text{prime } p \text{ for } p\)
proof -
have \(\sum a. k. -(\text{fds nth } f p \div \text{nat-power } p s) \div \text{Suc } k \div \text{of-real } (\ln (\text{real } p))) = (\sum a. k. (\text{fds nth } f p \div \text{nat-power } p s) \div \text{Suc } k) \div \text{of-real } (\ln (\text{real } p))\)
using \(\text{sum4[of } p]\)
by \((\text{subst infsetsum-cmult-left [symmetric]}\)
\((\text{auto simp: abs-summable-on-nat-iff' norm-power simp del: power-Suc})\)
also have \(\sum a. k. (\text{fds nth } f p \div \text{nat-power } p s) \div \text{Suc } k) = \ldots \ldots\)
\[ \frac{1}{1 - \text{fds-nth } f \ p / \text{nat-power } p \ s} - 1 \]

Using \text{sum4[OF p]}

\text{sums[OF p]}

by (subst infsetsum-nat')

(auto simp: sums-iff abs-summable-on-nat-iff' norm-power simp del: power-Suc)

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

qed

\(\text{have sum3: } (\lambda x. \sum a y. -((\text{fds-nth } f x / \text{nat-power } x s) \sim \text{Suc } y \ast \text{of-real } (\ln (\text{real } x))))\)

abs-summable-on \{p. \text{prime } p\}

using \text{sum2} by (rule abs-summable-on-Sigma-project1') auto

also have \(\text{?this} \longleftrightarrow (\lambda p. -((\text{of-real } (\ln (\text{real } p))) \ast (1 / (1 - \text{fds-nth } f p / \text{nat-power } p s) - 1))))\) abs-summable-on \{p. \text{prime } p\}

by (intro abs-summable-on-cong eq auto)

also have \(\ldots \longleftrightarrow ?\text{th1} \) by (subst abs-summable-on-uminus-iff) auto

finally show \(?\text{th1} \).

\(\text{have eval-fds } h \ s = (\sum a n. \text{fds-nth } h n / \text{nat-power } n s)\)

using * unfolding eval-fds-def by (subst infsetsum-nat') auto

also have \(\ldots = (\sum a n \in \{n. \text{primepow } n\}. -\text{fds-nth } f n \ast \text{mangoldt } n / \text{nat-power } n s)\)

unfolding \(h\)-def using \text{fds-nth-logderiv-completely-multiplicative[OF assms(2)]}

(auto simp: \text{fds-nth-fds mantoldt-def} \text{aprimedivisor-prime-power ln-realpow prime-gt-0-nat}

\text{nat-power-power-left divide-simps simp del: power-Suc})

also have \(\ldots = (\sum a (p,k) \in (?P \times \text{UNIV}). -\text{fds-nth } f (p \sim \text{Suc } k) \ast \text{mangoldt } (p \sim \text{Suc } k) / \text{nat-power } (p \sim \text{Suc } k) s)\)

using \text{bij-betw-primepows unfolding case-prod-unfold}

by (intro infsetsum-reindex-bij-betw [symmetric])

also have \(\ldots = (\sum a (p,k) \in (?P \times \text{UNIV}). -((\text{fds-nth } f p / \text{nat-power } p s) \sim \text{Suc } k) \ast \text{of-real } (\ln (\text{real } p)))\)

by (intro infsetsum-cong)

(auto simp: \text{f.mult f.power mangoldt-def aprimedivisor-prime-power ln-realpow prime-gt-0-nat}

\text{nat-power-power-left divide-simps simp del: power-Suc})

also have \(\ldots = (\sum a p \mid \text{prime } p. \sum a k. -((\text{fds-nth } f p / \text{nat-power } p s) \sim \text{Suc } k) \ast \text{of-real } (\ln (\text{real } p)))\)

using \text{sum2} by (subst infsetsum-Times) (auto simp: \text{case-prod-unfold})

also have \(\ldots = (\sum a p \mid \text{prime } p. -((\text{of-real } (\ln (\text{real } p))) \ast (1 / (1 - \text{fds-nth } f p / \text{nat-power } p s) - 1))))\)

using \text{eq} by (intro infsetsum-cong) auto

finally show \(?\text{th2} \) by (subst \text{asm} infsetsum-uminus)

qed

\text{lemma eval-fds-logderiv-zeta:}

assumes \(\text{Re } s > 1\)

shows \((\lambda p. \text{of-real } (\ln (\text{real } p)) / (p \text{ powr } s - 1))\)

abs-summable-on \{p. \text{prime } p\} (is \ ?\text{th1})
and deriv zeta s / zeta s =
\[-(\sum_{p | \text{prime } p} \ln (\text{real } p) / (p \text{ powr } s - 1))\] (is ?th2)

proof –
have *: completely-multiplicative-function (fds-nth fds-zeta :: - ⇒ complex)
by standard auto

note abscissa = le-less-trans[OF abs-conv-abscissa-completely-multiplicative-log-deriv[OF *]]

have \((\lambda p. \ln (\text{real } p) \ast (1 / (1 - \text{fds-nth } fds\text{-zeta } p / p \text{ powr } s - 1)))\) abs-summable-on \(\{p. \text{prime } p\}\)
using eval-fds-logderiv-completely-multiplicative[OF * of s] assms by auto

also have \(?this \iff (\lambda p. \ln (\text{real } p) / (p \text{ powr } s - 1))\) abs-summable-on \(\{p. \text{prime } p\}\)
by (intro abs-summable-on-cong) (auto simp: fds-nth-zeta divide-simps dest: prime-gt-1-nat)

finally show ?th1.

qed

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
proof –

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

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

qed

lemma range-add-nat: range \((\lambda n. n + c) = \{(c::nat)\}\)
using Nat.le-imp-diff-is-add by auto

lemma abs-summable-hurwitz-zeta:
assumes \( \text{Re } s > 1 \text{ and } \text{Re } b > 0 \)
shows \( (\lambda n. 1 / (\text{of-nat } n + a) \text{ powr } s) \text{ abs-summable-on } \{b.\} \)
proof
from assms have summable \((\lambda n. \text{cmod } (1 / (\text{of-nat } (n + b) + a) \text{ powr } s))\)
  using summable-hurwitz-zeta-real[of \( \text{Re } s a + b \)]
  by (auto simp: norm-divide powr-minus field-simps norm-powr-real-powr)
hence \((\lambda n. 1 / (\text{of-nat } (n + b) + a) \text{ powr } s) \text{ abs-summable-on } UNIV\)
  by (auto simp: abs-summable-on-nat-iff add-ac)
also have \(?this \longleftrightarrow (\lambda n. 1 / (\text{of-nat } n + a) \text{ powr } s) \text{ abs-summable-on range}\)
  \((\lambda n. n + b)\)
  by (rule abs-summable-on-reindex-iff) auto
also have range \((\lambda n. n + b) = \{b.\}\) by (rule range-add-nat)
finally show ?thesis .
qed

lemma hurwitz-zeta-nat-conv-infsetsum:
assumes \( a > 0 \text{ and } \text{Re } s > 1 \)
shows \( \text{hurwitz-zeta } (\lambda a \text{ real } a) s = \left(\sum_{a.n} \text{of-nat } (n + a) \text{ powr } s\right)\)
  \(\text{hurwitz-zeta } (\lambda a \text{ real } a) s = \left(\sum_{a.n\in\{a.\}} \text{of-nat } n \text{ powr } s\right)\)
proof
have \(\text{hurwitz-zeta } (\lambda a \text{ real } a) s = \left(\sum_{a.n} \text{of-nat } (n + a) \text{ powr } s\right)\)
  using assms by (subst hurwitz-zeta-conv-suminf) auto
also have \(\ldots = \left(\sum_{a.n} \text{of-nat } (n + a) \text{ powr } s\right)\)
  using assms by (intro infsetsum-nat’ [symmetric]) (auto simp: powr-minus field-simps)
finally show \(\text{hurwitz-zeta } (\lambda a \text{ real } a) s = \left(\sum_{a.n\in\text{range } (\lambda n. n + a). \text{of-nat } n \text{ powr } s}\right)\)
  by (rule infsetsum-reindex [symmetric]) auto
also have range \((\lambda a. n + a) = \{a.\}\) by (rule range-add-nat)
finally show \(\text{hurwitz-zeta } (\lambda a \text{ real } a) s = \left(\sum_{a.n\in\{a.\}} \text{of-nat } n \text{ powr } s\right)\).
qed

lemma pre-zeta-bound:
assumes \( 0 < \text{Re } s \text{ and } a: a > 0 \)
shows \( \text{norm } (\text{pre-zeta } a s) \leq (1 + \text{norm } s / \text{Re } s) / 2 * a \text{ powr } -\text{Re } s \)
proof
let \(?f = \lambda x. - (s * (x + a) \text{ powr } (-1-s))\)
let \(?g' = \lambda x. \text{norm } s * (x + a) \text{ powr } (-1-\text{Re } s)\)
let \(?g = \lambda x. -\text{norm } s / \text{Re } s * (x + a) \text{ powr } (-\text{Re } s)\)
define \(R\) where \(R = \text{EM-remainder } 1 \ ?f 0\)
have \([\text{simp}]: -\text{Re } s - 1 = -1 - \text{Re } s \text{ by } (\text{simp add: algebra-simps})\)
have \(|\text{frac } x - 1 / 2| \leq 1 / 2\) for \(x:: \text{real}\) unfolding frac-def
  by linarith
hence \(\text{pbernopoly } (\text{Suc } 0) \ x| \leq 1 / 2\) for \(x\)
  by (simp add: pbernopoly-def bernpoly-def)
moreover have \((\lambda b. \text{cmod } s * (b + a) \text{ powr } -\text{Re } s / \text{Re } s) \longrightarrow 0\) at-top
  using \(\text{Re } s > 0\) \(\langle a > 0\) by real-asym\)
ultimately have \(*: \forall x. x \geq \text{real } 0 \longrightarrow \text{norm } (\text{EM-remainder } 1 \ ?f (\text{int } x)) \leq\)
\( (1/2) \cdot \text{fact } 1 \ast (-?g \text{ (real } x)) \)

**lemma deriv-zeta-eq:**

```
assumes \( s : s \neq 1 \)
shows \( \text{deriv } \text{zeta } s = \text{deriv } (\text{pre-zeta } 1) \ast s = 1 \ast (s - 1)^2 \)
```

**proof –**

```
from \( s \) have \( \text{ev: eventually } (\lambda z. z \neq 1) \ast (\text{nhds } s) \) by \( \text{intro } \text{t1-space-nhds} \)
have \( \text{[derivative-intros]: (pre-zeta } 1 \text{ has-field-derivative deriv } (\text{pre-zeta } 1) \ast s) \) (at \( s \))
by \( \text{intro } \text{holomorphic-deriv}[\text{of } \text{UNIV}] \text{ holomorphic-intros} \) auto
have \( ((\lambda s. \text{pre-zeta } 1 \ast s + 1 / (s - 1)) \text{ has-field-derivative} \)
\( (\text{deriv } (\text{pre-zeta } 1) \ast s = 1 / (s - 1)^2)) \) (at \( s \))
using \( s \) by \( \text{auto intro!: derivative-eq-intros simp: power2-eq-square} \)
also have \( ?\text{this } \leftrightarrow (\text{zeta has-field-derivative deriv } (\text{pre-zeta } 1) \ast s = 1 / (s - 1)^2)) \) (at \( s \))
by \( \text{intro } \text{has-field-derivative-cong-ev eventually-monoi}[\text{OF } \text{ev}] \)
(auto simp: \text{zeta-def hurwitz-zeta-def})
```

**qed

lemma pre-zeta-bound':

```
assumes \( 0 < \text{Re } s \text{ and } a : a > 0 \)
shows \( \text{norm } (\text{pre-zeta } a \ast s) \leq \text{norm } s / (\text{Re } s \ast a \text{ powr } \text{Re } s) \)
```

**proof –**

```
from \( \text{assms} \) have \( \text{norm } (\text{pre-zeta } a \ast s) \leq (1 + \text{norm } s / \text{Re } s) / 2 \ast a \text{ powr } \text{Re } s \)
by \( \text{(intro pre-zeta-bound) auto} \)
also have \( \ldots = (\text{Re } s + \text{norm } s) / 2 / (\text{Re } s \ast a \text{ powr } \text{Re } s) \)
using \( \text{assms} \) by \( \text{(auto simp: field-simps powr-minus)} \)
also have \( \text{Re } s + \text{norm } s \leq \text{norm } s + \text{norm } s \) by \( \text{(intro add-right-mono complex-Re-le-cmod} \)
also have \( \text{norm } s + \text{norm } s / 2 = \text{norm } s \) by \( \text{simp} \)
finally show \( \text{norm } (\text{pre-zeta } a \ast s) \leq \text{norm } s / (\text{Re } s \ast a \text{ powr } \text{Re } s) \)
using \( \text{assms} \) by \( \text{(simp add: divide-right-mono)} \)
```

**qed

```
```
\[
\text{finally show } \theta \text{ by } \text{(rule DERIV-imp-deriv)}
\]
\text{qed}

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

lemma \text{eval-fds-deriv-zeta:}
\begin{align*}
&\text{assumes } \Re s > 1 \\
&\text{shows } \text{eval-fds (fds-deriv fds-zeta) s} = \text{deriv zeta s}
\end{align*}
\begin{align*}
&\text{proof -- } \\
&\text{have } \text{ev: eventually } (\lambda z. z \in \{z. \Re z > 1\}) \text{ (nhds s)} \\
&\text{using assms by } \text{(intro eventually-nhds-in-open open-halfspace-Re-gt)} \\
&\text{from assms have } \text{eval-fds (fds-deriv fds-zeta) s} = \text{deriv (eval-fds fds-zeta) s} \\
&\text{by } \text{(subst eval-fds-deriv)} \\
&\text{also have } \ldots = \text{deriv zeta s} \\
&\text{by } \text{(intro deriv-cong-ev eventually-mono[OF ev]) (auto simp: eval-fds-zeta)} \\
&\text{finally show } \theta.
\end{align*}
\text{qed auto}

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

lemma \text{sum-upto-plus1:}
\begin{align*}
&\text{assumes } x \geq 0 \\
&\text{shows } \text{sum-upto f (x + 1) = sum-upto f x + f (Suc (nat [x])})
\end{align*}
\begin{align*}
&\text{proof -- } \\
&\text{have } \text{sum-upto f (x + 1) = sum f \{0<..<Suc (nat [x])\}} \\
&\text{using assms by } \text{(simp add: sum-upto-altdef nat-add-distrib)} \\
&\text{also have } \text{\{0<..<Suc (nat [x])\} = insert (Suc (nat [x])) \{0<..nat [x]\}} \\
&\text{by auto} \\
&\text{also have } \text{sum f \ldots = sum-upto f x + f (Suc (nat [x])}} \\
&\text{by } \text{(subst sum.insert) (auto simp: sum-upto-altdef add-ac)} \\
&\text{finally show } \theta.
\end{align*}
\text{qed}

lemma \text{sum-upto-minus1:}
\begin{align*}
&\text{assumes } x \geq 1 \\
&\text{shows } \text{sum-upto f (x - 1) = (sum-upto f x - f (nat [x]) :: 'a :: ab-group-add)} \\
&\text{using sum-upto-plus1[of x - 1 f] assms by } \text{(simp add: algebra-simps nat-diff-distrib)}
\end{align*}

lemma \text{integral-smallo:}
\begin{align*}
&\end{align*}

28
\textbf{proof} (rule landau-o.smallI)

\begin{itemize}
\item \textbf{fix} \( c :: \text{real} \Rightarrow \text{real} \)
\item \textbf{assume} \( c > 0 \)
\item \textbf{define} \( c' \text{ where } c' = c / 2 \)
\item \textbf{from} \( c \) \textbf{have} \( c' \cdot c' > 0 \) by (simp add: \( c'\text{-def} \))
\item \textbf{from} landau-o.smallI[OF assms(1) this]
\item \textbf{obtain} \( b \text{ where } b : \forall x. x \geq b \Rightarrow \text{norm} (f x) \leq c' \cdot \text{norm} (g' x) \)
\item \textbf{unfolding} eventually-at-top-linorder by blast
\item \textbf{define} \( b' \text{ where } b' = \max a b \)
\item \textbf{define} \( D \text{ where } D = \text{norm} (\text{integral} \{a..b'\} f) \)
\end{itemize}

\textbf{have} \( \text{filterlim} (\lambda x. c' \cdot g x) \) \text{at-top} at-top
\textbf{using} \( c' \) by (intro filterlim-tendsto-pos-mult-at-top[OF tendsto-const] assms)

\textbf{hence} \( \text{eventually} (\lambda x. c' \cdot g x \geq D - c' \cdot g b') \text{at-top} \)
\textbf{by} (auto simp: filterlim-at-top)

\textbf{thus} \( \text{eventually} (\lambda x. \text{norm} (\text{integral} \{a..x\} f) \leq c \cdot \text{norm} (g x)) \) \text{at-top} by
\textbf{eventually-ge-at-top[of } b'\]

\textbf{proof} \textbf{eventually-elim}
\textbf{case} \( (\text{elim } x) \)
\textbf{have} \( b': a \leq b' \leq b' \) \textbf{by} (auto simp: b'-def)
\textbf{from} \( \text{elim } b' \) \textbf{have} \( \text{integrable'} (\lambda x. |g' x|) \text{integrable-on } \{b'..x\} \)
\textbf{by} (intro integrable-continuous-real continuous-intros auto)
\textbf{have} \( \text{integral } \{a..x\} f = \text{integral } \{a..b\} f + \text{integral } \{b'..x\} f \)
\textbf{using} \( \text{elim } b' \text{by} \) (intro Henstock-Kurzweil-Integration.integral-combine [symmetric]
\textbf{assms} auto)
\textbf{also have} \( \text{norm } \ldots \leq D + \text{norm} (\text{integral } \{b'..x\} f) \)
\textbf{unfolding} \( D\text{-def by} \) (rule norm-triangle-ineq)
\textbf{also have} \( \text{norm} (\text{integral } \{b'..x\} f) \leq \text{integral } \{b'..x\} (\lambda x. c' \cdot \text{norm} (g' x)) \)
\textbf{using} \( b' \text{ elim assms } c' \text{ integrable by} \) (intro integral-bound-continuous b
\textbf{assms} auto)
\textbf{also have} \( \ldots = c' \cdot \text{integral } \{b'..x\} (\lambda x. |g' x|) \) \textbf{by simp}
\textbf{also have} \( \text{integral } \{b'..x\} (\lambda x. |g' x|) = \text{integral } \{b'..x\} g' \)
\textbf{using} \( \text{assms } b' \text{ by} \) (intro integral-cong) auto
\textbf{also have} \( (g' \text{ has-integral } (g x - g b') \) \{b'..x\} \text{ using } b' \text{ elim}
\textbf{by} (intro fundamental-theorem-of-calculus)
\text{auto simp flip: has-real-derivative-iff-has-vector-derivative}
\text{intro! has-field-derivative-at-within[OF deriv]})
\textbf{hence} \( \text{integral } \{b'..x\} g' = g x - g b' \)
\textbf{by} (simp add: has-integral-iff)
\textbf{also have} \( D + c' \cdot (g x - g b') \leq c \cdot g x \)
\textbf{using} \( \text{elim by} \) (simp add: field-simps c'-def)
\textbf{also have} \( \ldots \leq c \cdot \text{norm} (g x) \)
proof

eventually hence eventually 

filterlim have have D define b define c define

landau-o note \([\lambda x. (\text{integral } \{a..x\}) f] \in O(g)\)

proof

- note \([\text{continuous-intros} = \text{continuous-on-subset}[\text{OF cont}]\]

obtain c b where c: c > 0 and b: \(\forall x. \ x \geq b \implies \|f(x)\| \leq c \ast \|g'(x)\|\)

unfolding eventually-at-top-linorder by metis
define c' where c' = c / 2
define b' where b' = max a b
define D where D = norm (integral \(\{a..b'\}\) f)

have filterlim (\(\lambda x. \ c \ast g(x)\)) at-top at-top

using c by (intro filterlim-tends-to-pos-mult-at-top[OF tends-to-const] assms)
hence eventually (\(\lambda x. \ c \ast g(x) \geq D - c \ast g(b')\)) at-top

by (auto simp: filterlim-at-top)
hence eventually (\(\lambda x. \ \text{norm} (\text{integral } \{a..x\}) f \leq 2 \ast c \ast \text{norm} (g(x))\)) at-top

using eventually-ge-at-top[of b']

proof eventually-elim
case (elim x)

have b': a \leq b' b \leq b' by (auto simp: b'-def)

from elim b' have integrable: (\(\lambda x. \ |g'(x)|\)) integrable-on \(\{b'..x\}\)

by (intro integrable-continuous-real continuous-intros auto)

have integral \(\{a..x\}\) f = integral \(\{a..b'\}\) f + integral \(\{b'..x\}\) f

using elim b' by (intro Henstock-Kurzweil-Integration.integral-combine [symmetric] assms) auto

also have \(\|\|\|f\|\| \leq D + \text{norm} (\text{integral } \{b'..x\}) f\)

unfolding D-def by (rule norm-triangle-ineq)
also have \(\text{norm} (\text{integral } \{b'..x\}) f \leq \text{integral} \{b'..x\} (\lambda x. \ c \ast \text{norm} (g'(x)))\)

using b' elim assms c integrable by (intro integral-norm-bound-integral b assms) auto

also have \(\ldots = c \ast \text{integral} \{b'..x\} (\lambda x. \ |g'(x)|)\) by simp
also have integral \(\{b'..x\}\) (\(\lambda x. \ |g'(x)|\)) = integral \(\{b'..x\}\) g'

using assms b' by (intro integral-cong) auto
also have (\(g'\) has-integral \(g \ast g'(b')\)) \(\{b'..x\}\) using b' elim

by (intro fundamental-theorem-of-calculus)
\((\text{auto simp flip; has-real-derivative-iff-has-vector-derivative intro!: DERIV-subset[OF deriv]}\))

qed

lemma integral-bigo:

fixes f g g' :: real \Rightarrow real
assumes f \in O(g) and filterlim g at-top at-top
assumes \(\lambda a'. x. \ a \leq a' \implies a' \leq x \implies f \text{integrable-on } \{a'..x\}\)
assumes deriv: \(\lambda x. \ x \geq a \implies (g \text{ has-field-derivative } g'(x))(\text{at } x \text{ within } \{a..\})\)
assumes cont: continuous-on \(\{a..\}\) g'
assumes nonneg: \(\lambda x. \ x \geq a \implies g'(x) \geq 0\)
shows (\(\lambda x. \ \text{integral } \{a..x\} f \in O(g)\))

Finally show \(?\text{case by simp}\)

qed
hence \[
\int b' \cdot x \, g' = g \, x - g \, b' \]
by (simp add: has-integral-iff)
also have \[
D + c \cdot (g \, x - g \, b') \leq 2 \cdot c \cdot g \, x
\]
using elim by (simp add: field-simps c'-def)
also have \[
\ldots \leq 2 \cdot c \cdot \text{norm}(g \, x)
\]
using c by (intro mult-left-mono) auto
finally show \textquote{case} by simp
qed
thus \textquote{thesis} by (rule bigoI)
qed

lemma primepows-le-subset:
assumes \[
x > 0 \text{ and } l > 0
\]
shows \[
\{(p, i) \mid \text{prime } p \land l \leq i \land \text{real}(p \, ^i) \leq x\} \subseteq \{\text{nat } \lfloor \text{root } l \, x \rfloor\} \times \{\text{nat } \lfloor \log 2 \, x \rfloor\}
\]
proof safe
fix \(p \, i\) :: nat
assume \[
pi : \text{prime } p \land i \geq l \land \text{real}(p \, ^i) \leq x
\]
have \[
\text{real } p \, ^i \leq \text{real } p \, ^l
\]
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 \[
\text{p} \leq \text{nat } \lfloor \text{root } l \, x \rfloor
\]
using \(pi \, l \, x\) by (simp add: le-nat-iff le-floor-iff le-log-iff powr-realpow)
qed

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

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 1])
fix \(x\) :: real
assume \[
x \geq 1
\]
from \(x\) have \[
* : x - \text{real } (\lfloor x \rfloor) \leq 1
\]
by linarith
from \(x\) have \[
\ln x - \ln (\text{real } (\lfloor x \rfloor)) \leq (x - \text{real } (\lfloor x \rfloor)) / \text{real } (\lfloor x \rfloor)
\]
by (intro ln-diff-le) auto
also have \[
\ldots \leq 1 / 1
\]
using \(x \, *\) by (intro frac-le) auto
finally show \[
\ln x - \ln (\text{real } (\lfloor x \rfloor)) \leq 1 * 1
\]
by simp
qed auto

lemma cos-geD:
assumes \[
\cos x \geq \cos a \quad 0 \leq a \leq a \quad \neg pi \quad -pi \leq x \leq pi
\]
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 \(-a\)]
    by auto
qed

lemma path-image-part-circlepath-same-Re:
  assumes \( 0 \leq b \leq \pi \) \( a = -b \) \( r \geq 0 \)
  shows \( \text{path-image} (\text{part-circlepath} c r a b) = \text{sphere} c r \cap \{ s. \text{Re} s \geq \text{Re} c + \text{r} \ast \text{cos} a \} \)
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 assms show \( \text{Re} z \geq \text{Re} c + \text{r} \ast \text{cos} a \)
    using cos-monotone-0-pi-le[of \( t \)] cos-monotone-minus-pi-0[of \( a \)]
    by (cases \( t \geq 0 \)) (auto intro: mult-left-mono)
next
  fix \( z \) assume \( z: z \in \text{sphere} c r \) \( \text{Re} z \geq \text{Re} c + \text{r} \ast \text{cos} a \)
  show \( z \in \text{path-image} (\text{part-circlepath} c r a b) \)
    proof (cases \( r = 0 \))
      case False
      with assms have \( r: r > 0 \) by simp
      with \( z \) have \( z\text{-eq}: z = c + r \ast \text{cis} (\text{Arg} (z - c)) \)
        using Arg-eq[of \( z - c \)] by (auto simp: dist-norm exp-eq-polar norm-minus-commute)
      moreover from \( z(2) \) r assms have \( \text{cos} b \leq \text{cos} (\text{Arg} (z - c)) \)
        by (subst (asm) \( z\text{-eq} \)) auto
      with assms have \( \text{Arg} (z - c) \in \{-b..b\} \)
        using Arg-le-pi[of \( z - c \)] mpi-less-Arg[of \( z - c \)] by (intro cos-geD) auto
      ultimately show \( z \in \text{path-image} (\text{part-circlepath} c r a b) \)
        using assms by (subst path-image-part-circlepath) (auto simp: exp-eq-polar)
    qed (insert assms \( z \), auto simp: path-image-part-circlepath)
qed

lemma part-circlepath-rotate-left:
  \( \text{part-circlepath} c r (x + a) (x + b) = (\lambda z. c + \text{cis} x \ast (z - c)) \circ \text{part-circlepath} c r a b \)
  by (simp add: part-circlepath-def exp-eq-polar fun-eq-iff
  linepath-translate-left linepath-translate-right cis-mult add-ac)

lemma 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: part-circlepath-def exp-eq-polar fun-eq-iff linepath-translate-left linepath-translate-right cis-mult add-ac)

**Lemma** path-image-semicircle-Re-ge:
assumes \( r \geq 0 \)
shows \( \text{path-image (part-circlepath } c \ r \ (-\pi/2) \ (\pi/2)) = \text{sphere } c \ r \cap \{ s. \ \Re s \geq \Re c \} \)
by (subst path-image-part-circlepath-same-Re) (simp-all add: assms)

**Lemma** sphere-rotate: \((\lambda z. \ c + \text{cis } x * (z - c)) \ ' \ \text{sphere } c \ r = \text{sphere } c \ r \)
proof safe
fix \( z \) assume \( z \in \text{sphere } c \ r \)

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: dist-norm norm-mult norm-minus-commute cis-conv-exp exp-minus field-simps norm-divide)
with \( z \) show \( z \in (\lambda z. \ c + \text{cis } x * (z - c)) \ ' \ \text{sphere } c \ r \) by blast
qed (auto simp: dist-norm norm-minus-commute norm-mult)

**Lemma** path-image-semicircle-Re-le:
assumes \( r \geq 0 \)
shows \( \text{path-image (part-circlepath } c \ r \ (0 \ \pi) \ (3/2*\pi)) = \text{sphere } c \ r \cap \{ s. \ \Re s \leq \Re c \} \)
proof
let \( ?f = (\lambda z. \ c + \text{cis } \pi/2 * (z - c)) \)
have \( \ast: \text{part-circlepath } c \ r \ (\pi/2) \ (3/2*\pi) = \text{part-circlepath } c \ r \ (\pi + (-\pi/2)) \ (\pi + \pi/2) \)
by simp
have \( \text{path-image (part-circlepath } c \ r \ (\pi/2) \ (3/2*\pi)) = ?f ' \ \text{sphere } c \ r \cap ?f ' \ \{ s. \ \Re c \leq \Re s \} \)
unfolding \ast path-image-compose path-image-semicircle-Re-ge[OF assms]
by auto
also have \( ?f ' \ \text{sphere } c \ r = \text{sphere } c \ r \)
by (rule sphere-rotate)
also have \( ?f ' \ \{ s. \ \Re c \leq \Re s \} = \{ s. \ \Re c \geq \Re s \} \)
by (auto simp: image-iff intro!: exI[of - 2 * c - x for x])
finally show \( ?\text{thesis} \).
qed

**Lemma** path-image-semicircle-Im-ge:
assumes \( r \geq 0 \)
shows \( \text{path-image (part-circlepath } c \ r \ 0 \ \pi) = \text{sphere } c \ r \cap \{ s. \ \Im s \geq \Im c \} \)
proof
let \( ?f = (\lambda z. \ c + \text{cis } (\pi/2) * (z - c)) \)
have \( \ast: \text{part-circlepath } c \ r \ 0 \ \pi = \text{part-circlepath } c \ r \ (\pi / 2 + (-\pi/2)) (\pi / 2 + \pi/2) \) by simp

have \( \text{path-image } (\text{part-circlepath } c \ r \ 0 \ \pi) = \)
\( \notin' \ \text{sphere } c \ r \ \cap \ \notin' \ \{s. \ Re c \leq Re s\} \)

unfolding \( \ast \ \text{part-circlepath-rotate-left path-image-compose path-image-semicircle-Re-ge}\)

by simp

have \( \text{path-image } (\text{part-circlepath } c \ r \ 0 \ \pi) = \)
\( \notin' \ \text{sphere } c \ r \ \cap \ \notin' \ \{s. \ Re c \leq Re s\} \)

unfolding \( \ast \ \text{part-circlepath-rotate-left path-image-compose path-image-semicircle-Re-ge}\)

by auto

also have \( \notin' \ \text{sphere } c \ r = \text{sphere } c \ r \)
by \( \text{rule sphere-rotate} \)

also have \( \notin' \ \{s. \ Re c \leq Re s\} = \{s. \ Im c \leq Im s\} \)

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

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

qed

lemma \( \text{path-image-semicircle-Im-le} \):

assumes \( r \geq 0 \)

shows \( \text{path-image } (\text{part-circlepath } c \ r \ (2 * \pi)) = \text{sphere } c \ r \ \cap \ \{s. \ Im s \leq Im c\} \)

proof –

let \( \notin = (\lambda z. \ c + \text{cis } (3 * \pi/2) * (z - c)) \)

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

unfolding \( \ast \ \text{part-circlepath-rotate-left path-image-compose path-image-semicircle-Re-ge}\)

by auto

also have \( \notin' \ \text{sphere } c \ r = \text{sphere } c \ r \)
by \( \text{rule sphere-rotate} \)

also have \( \text{cis } (3 * \pi / 2) = -i \)
using \( \text{cis-mult}[\text{of pi pi} / 2] \) by simp

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

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

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

qed

lemma \( \text{eval-fds-logderiv-zeta-real} \):

assumes \( x > (1 :: \text{real}) \)

shows \( (\lambda p. \ \ln (\text{real } p) / (p \ \text{pwr } x - 1)) \ \text{abs-summable-on } \{p. \ \text{prime } p\} \) (is \( ?\text{th1} \))

and \( \text{deriv zeta } (\text{of-real } x) / \ zeta (\text{of-real } x) = \)
\( -\text{of-real } (\sum p \ | \ \text{prime } p. \ \ln (\text{real } p) / (p \ \text{pwr } x - 1)) \) (is \( ?\text{th2} \))

proof –

have \( (\lambda p. \ Re (\text{of-real } (\ln (\text{real } p)) / \text{of-nat } p \ \text{pwr } \text{of-real } x - 1))) \)
\( \text{abs-summable-on } \{p. \ \text{prime } p\} \) using \( \text{assms} \)

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

also have \( \text{thesis} \longleftrightarrow ?\text{th1} \)

34
lemma

fixes a b c d :: real
assumes ab: (d * a + b) ≤ 1 and c: (c < −1) and d: d > 0
defines C ≡ − (((ln (d * a + b) − 1) / (c + 1)) * (d * a + b) / (d * (c + 1))))
shows (∀ x ∈ {a < ..}. (ln (d * x + b) * ((d * x + b) powr c)) has-integral C) {a < ..} (is ?th1)
and (∀ x ∈ {a < ..}. (ln (d * x + b) * ((d * x + b) powr c)) has-integral C) {a < ..} (is ?th2)

proof –

define f where f = (λx. ln (d * x + b) * ((d * x + b) powr c))
define F where F = (λx. (ln (d * x + b) − 1) / (c + 1)) * (d * x + b) powr (c + 1) / (d * (c + 1)))

have *: (F has-field-derivative f x) (at x) isCont f x f x ≥ 0 if x > a for x

proof –

have 1 ≤ d * a + b by fact
also have ⟹ < d * x + b using that assms
by (intro add-strict-right-mono mult-strict-left-mono)
finally have gt-1: d * x + b > 1.
show (F has-field-derivative f x) (at x) isCont f x using ab c d gt-1
by (auto simp: F-def f-def divide-simps intro!: derivable-eq-intros continuous-intros)
(auto simp: algebra-simps powr-add)!
show f x ≥ 0 using gt-1 by (auto simp: f-def)

qed
show \[ \text{th3} \]
using set-borel-integral-eq-integral[OF 1] 3 by (simp add: has-integral-iff f-def C-def)
qed

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

lemma sum-upto-ln-conv-ln-fact: \( \text{sum-upto } \ln(x) = \ln(\text{fact } \lceil x \rceil) \)
by (simp add: ln-fact-conv-sum-upto sum-upto-altdef)

lemma real-of-nat-div: \( \text{real}(a \div b) = \text{real}(\text{int} (a \div b)) \)
by (simp add: floor-divide-of-nat-eq)

lemma measurable-sum-upto [measurable]:
fixes \( f : 'a \Rightarrow \text{nat} \Rightarrow \text{real} \)
assumes [measurable]: \( \forall y. (\lambda t. f t y) \in M \rightarrow M \)
borel
assumes [measurable]: \( x \in M \rightarrow M \)
borel
shows \( (\lambda t. \text{sum-upto } (f t)(x t)) \in M \rightarrow M \)
borel
proof
  have meas: \( (\lambda t. \text{set-lebesgue-integral } \{ y. y \geq 0 \land y - \text{real } (\text{nat } x t) \leq 0 \} ) (\lambda y. f t (\text{nat } \lceil y \rceil)) \)
    \in M \rightarrow M borel (is ?f ∈ -) 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 (cases \( x t < 1 \))
case True
  hence \( \{ y. y \geq 0 \land y - \text{real } (\text{nat } x t) \leq 0 \} = \{ 0 \} \) by auto
  thus ?thesis using True
  by (simp add: set-integral-at-point sum-upto-altdef)
next
case False
define \( n \) where \( n = \text{nat } \lfloor x t \rfloor \)
from False have \( n > 0 \) by (auto simp: n-def)

have \( *: ((\lambda x. f t (\text{nat } \lfloor x t \rfloor)) \text{ has-integral sum } (f t) \{ 0 <..n \}) \{ \text{real } 0\ldots \text{real } n \} \)
  using \( n > 0 \) by (intro nat-sum-has-integral-ceiling) auto

have \( **: ((\lambda x. f t (\text{nat } \lfloor x t \rfloor)) \text{ absolutely-integrable-on } \{ \text{real } 0\ldots \text{real } n \} \)
  proof (rule absolutely-integrable- absolutely-integrable-ubound)
    show \( (\lambda x. f t (\text{nat } \lfloor x t \rfloor)) \text{ absolutely-integrable-on } \{ \text{real } 0\ldots \text{real } n \} \)
      using \( n > 0 \) by (subst absolutely-integrable-on-iff-nonneg)
        (auto simp: Max-ge-iff intro: \( \exists! f t \) 0)
    show \( (\lambda x. f t (\text{nat } \lfloor x t \rfloor)) \text{ integrable-on } \{ \text{real } 0\ldots \text{real } n \} \)
      using \( * \) by (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 neumman-ingham-aux1:
  fixes $R :: real$ and $z :: complex$
  assumes $R : R > 0$ and $z : norm z = R$
  shows $\|1 / z + z / R^2\| = 2 * |Re z| / R^2$
  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:**

\[
\text{by (simp add: field-simps power2-eq-square)}
\]

also have \( \text{norm } R^2 + z^2 = z * (z + \text{cnj } z) \) using complex-norm-square[of \( z \)]

\[
\text{by (simp add: z power2-eq-square algebra-simps)}
\]

also have \( \text{norm } R^2 = 2 * |\text{Re } z| * R \)

\[
\text{by (subst complex-add-cnj) (simp-all add: z norm-mult)}
\]

also have \( R \) by (simp add: field-simps numeral-3-eq-3 power2-eq-square)

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

qed

**lemma neuman-ingham-aux2:**

\[
\text{fixes } m :: \text{nat and } w z :: \text{complex}
\]

\[
\text{assumes } 1 \leq m \leq \text{Re } w < \text{Re } z \text{ and } f; \forall n. 1 \leq n \Rightarrow \text{norm } (f n) \leq C
\]

\[
\text{shows } \text{norm } (\sum n=1..m. f n / \text{n powr } (w - z)) \leq C * (m \text{ powr } \text{Re } z) * (1 / m + 1 / \text{Re } z)
\]

**proof**

\[
\text{have } [\text{simp}]: C \geq 0 \text{ by (rule order.trans[OF - f[of 1]]) auto}
\]

\[
\text{have } \text{norm } (\sum n=1..m. f n / \text{n powr } (w - z)) \leq (\sum n=1..m. C / \text{n powr } (1 - \text{Re } z))
\]

\[
\text{by (rule sum-norm-le)}
\]

\[
(\text{insert assms, auto simp: norm-divide norm-powr-real-powr intro!: frac-le assms powr-mono})
\]

also have \( \dots = C * (\sum n=1..m. n \text{ powr } (\text{Re } z - 1)) \)

\[
\text{by (subst sum-distrib-left) (simp-all add: powr-diff)}
\]

also have \( \dots \leq C * (m \text{ powr } \text{Re } z * (1 / \text{Re } z + 1 / m)) \)

\[
\text{using zeta-partial-sum-le[of Re z m] assms by (intro mult-left-mono) auto}
\]

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

qed

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

\[
\text{fixes } a x :: \text{real}
\]

\[
\text{assumes } \text{ax: } a > 0 \text{ x > 1}
\]

\[
\text{shows } (\sum i. (a + \text{real } \text{Suc } i) \text{ powr } (-x)) \leq a \text{ powr } (1 - x) / (x - 1)
\]

**proof** (rule decreasing-sum-le-integral, goal-cases)

\[
\text{have } ((\lambda t. (a + t) \text{ powr } -x) \text{ has-integral } -(a \text{ powr } (-x + 1)) / (-x + 1))
\]

\[
(\text{interior } \{0..\})
\]

using powr-has-integral-at-top[of 0 a -x] using \text{ax} by (simp add: interior-real-atLeast)

also have \( -(a \text{ powr } (-x + 1)) / (-x + 1) = a \text{ powr } (1 - x) / (x - 1) \)

using \text{ax} by (simp add: field-simps)

finally show \( ((\lambda t. (a + t) \text{ powr } -x) \text{ has-integral } a \text{ powr } (1 - x) / (x - 1)) \)

\[
\{0..\}
\]

\[
\text{by (subst (asm) has-integral-interior) auto}
\]

qed (insert \text{ax}, auto intro!: powr-mono2')
fixes y1 y2 x :: real
defines $S \equiv \{ z \in \mathbb{C} \mid \text{Re } z = x \wedge \text{Im } z \in [y1..y2] \}$
assumes y1 \leq y2
assumes f analytic-on $S$
obtains x1 x2 :: real where $x1 < x x2 > x$ analytic-on cbox (Complex x1 y1) (Complex x2 y2)
proof –
define C where $C = \{ \text{box } a b \mid a b z \in S \}
\text{have } S = \text{cbox } (\text{Complex } x y1) (\text{Complex } x y2)
\text{by (auto simp: S-def in-cbox-complex-iff)}
also have compact ... by simp
finally have 1: compact $S$.

have 2: $S \subseteq \bigcup C$
proof (intro subsetI)
fix z assume z \in S
from f analytic-on $S$ and this obtain a b where $z \in \text{box } a b$ f analytic-on box a b
by (blast elim: analytic-onE-box)
with \{z \in S\} show $z \in \bigcup C$ unfolding C-def by blast
qed

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 $T \subseteq \bigcup T$
by blast
define x1 where $x1 = \text{Max } \{ x - 1 \} (\lambda X. x + (\text{Re } X - x) / 2 \bullet T))$
define x2 where $x2 = \text{Min } \{ x + 1 \} (\lambda X. x + (\text{Sup } X - x) / 2 \bullet T))$

have \*: x + (\text{Inf } (\text{Re } X - x) / 2 < x \wedge x + (\text{Sup } (\text{Re } X - x) / 2 > x \text{ if } X \in T \text{ for } X$
proof –
from that and T obtain a b s where [simp]: $X = \text{box } a b$ and s: $s \in \text{box } a b$
s \in S
by (force simp: C-def)
\text{hence le: Re a < Re b Im a < Im b by (auto simp: in-box-complex-iff)}
show \?thesis using le s
unfolding \{X = \text{box } a b \} Re-image-box[OF le] Im-image-box[OF le]
by (auto simp: S-def in-box-complex-iff)
qed
from \* T have \{x1 < x unfolding x1-def by (subst Max-less-iff) auto
from \* T have \{x2 > x unfolding x2-def by (subst Min-gr-iff) auto

have f analytic-on $\bigcup T$
using T by (subst analytic-on-Union) (auto simp: C-def)
moreover have \{z \in \bigcup T \text{ if } z \in \text{cbox } (\text{Complex } x1 y1) (\text{Complex } x2 y2) \text{ for } z

39
proof –

from that have Complex x (Im z) ∈ S
  by (auto simp: in-cbox-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 ≤ x2 by (simp add: in-cbox-complex-iff)
also have … ≤ x + (Sup (Re ' X) − x) / 2
  unfolding x2-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-cbox-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 cbox (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.

** lemma Newman-Ingham-1:**

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

**proof**

- We get a bound on our coefficients and call it \( C \).
- **have** \( (\lambda N. \text{eval-fds} (\text{fds-truncate} N F) w) \rightarrow f w \)
- **unfolding** \( \text{tendsto-iff dist-norm norm-minus-commute[of eval-fds} F s \text{ for } F s \)

**proof**

- **fix** \( \varepsilon : \text{real} \)
- **assume** \( \varepsilon : \varepsilon > 0 \)
- We choose an integration radius that is big enough for the error to be sufficiently small.
- **define** \( R \) where \( R = \max 1 (3 \ast C / \varepsilon) \)
- **have** \( R \geq 3 \ast C / \varepsilon \)
- **define** \( l \) where \( l > 0 \)
- \( (\lambda z. f (w + z)) \text{ analytic-on} \{ s, \text{Re} \ s > 0 \lor \text{Im} s \in \{-R-1..<R+1\} \land \text{Re} s > -l \} \)

  **proof**

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

  **by** \( \text{rule analytic-on-subset auto} \)

- **from** \( \text{analytic-on-axis-extend[OF - this]} \) **obtain** \( x1 \) \( x2 \) where \( x12 : \)
- \( x1 < 0 \land x2 > 0 \land (\lambda z. f (w + z)) \text{ analytic-on} \) \( \text{cbox} (\text{Complex} x1 (-R-1)) \)
- \( (\text{Complex} x2 (R+1)) \)
- **using** \( iR \geq 1 \) by auto
from this(3) have \((\lambda z. f (w + z))\) analytic-on \(\{ s. \Re s \in \{x1..0\} \land \Im s \in \{-R-1..R+1\}\}\) by (rule analytic-on-subset) (insert x12, auto simp: in-cbox-complex-iff)
with \(f\)-analytic' have \((\lambda z. f (w + z))\) analytic-on 
\(\{ s. \Re s \geq 0 \} \cup \{ s. \Re s \in \{x1..0\} \land \Im s \in \{-R-1..R+1\}\}\)
by (subst analytic-on-Un) auto
hence \((\lambda z. f (w + z))\) analytic-on \(\{ s. \Re s > 0 \lor \Im s \in \{-R-1..<R+1\}\}\)
\& \(\Re s > x1\)
by (rule analytic-on-subset) auto
with \(\langle x1 < \theta \rangle\) 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 \(X = box (Complex (-l) (-R-1)) (Complex (R+1) (R+1))\)
have [simp, intro]: open \(X\) convex \(X\) by (simp-all add: X-def open-box)
from \(R \lor l\) have [simp]: \(\emptyset \in X\) by (auto simp: X-def in-box-complex-iff)
have analytic: \((\lambda z. f (w + z))\) 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 \(\delta = \min (1/2) (l/2)\)
from \(l\) have \(\delta: \delta > 0 \land \delta \leq 1/2 \land \delta < l\) by (auto simp: \(\delta\)-def)
define \(X'\) where \(X' = box (Complex (-\delta) (-R)) (Complex R R)\)
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]: compact \(X'\) by (simp add: \(X'\)-def)
moreover have continuous-on \(X'\) (\(\lambda z. f (w + z)\))
using \(w \langle X' \subseteq X\rangle\) 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 \(\theta\) for \(N \to \infty\). We use the following bound:
define bound where
bound = (\lambda N::nat. \ ((2 \cdot C/R + C/N + 3 \cdot M / (pi \cdot R \cdot ln N) + 3 \cdot R \cdot M / (\delta \cdot pi \cdot N powr \delta))))

have \( 2 \cdot C / R < \varepsilon \) using \( M(1) \cdot R \cdot C(1) \cdot \delta(1) \cdot \varepsilon \)
  by (auto simp: field-simps)
  -- Evidently this is below \( \varepsilon \) for sufficiently large \( N \).

hence eventually \( (\lambda N::nat. \ bound N < \varepsilon) \) at-top
  using \( M(1) \cdot R \cdot C(1) \cdot \delta(1) \cdot \varepsilon \) unfolding bound-def
  by real-asymp

  -- It now only remains to show that the difference is indeed less than the claimed bound.
  thus eventually \( (\lambda N. \ norm (f \cdot w - eval-fds (fds-truncate N \cdot F) \cdot w) < \varepsilon) \) at-top
  using eventually-gt-at-top[of 1]
  proof eventually-elim
    case (elim \( N \))
    note \( N = this \)

    -- Like Harrison (and unlike Newman), our integration path \( \Gamma \) consists of a
    -- semicircle \( A \) of radius \( R \) in the right-halfplane and a box of width \( \delta \) and height \( 2R \)
    -- on the left halfplane. The latter consists of three straight lines, which we call \( B1 \) to \( B3 \).
    define \( A \) where \( A = part-circlepath \ 0 \ R \ (-pi/2) \ (pi/2) \)
    define \( B2 \) where \( B2 = linepath (Complex (-\delta) \ R) \ (Complex (-\delta) \ (-R)) \)
    define \( B1 \) where \( B1 = linepath (R \cdot i) \ (R \cdot i - \delta) \)
    define \( B3 \) where \( B3 = linepath (-R \cdot i - \delta) \ (-R \cdot i) \)
    define \( \Gamma \) where \( \Gamma = A +++ B1 +++ B2 +++ B3 \)

    -- We first need to show some basic facts about the geometry of our integration path.
    have [simp, intro]:
      path \( A \) path \( B1 \) path \( B3 \) path \( B2 \)
      valid-path \( A \) valid-path \( B1 \) valid-path \( B3 \) valid-path \( B2 \)
      arc \( A \) arc \( B1 \) arc \( B3 \) arc \( B2 \)
      pathstart \( A = -i \cdot R \) pathfinish \( A = i \cdot R \)
      pathstart \( B1 = i \cdot R \) pathfinish \( B1 = R \cdot i - \delta \)
      pathstart \( B3 = -R \cdot i - \delta \) pathfinish \( B3 = -i \cdot R \)
      pathstart \( B2 = R \cdot i - \delta \) pathfinish \( B2 = -R \cdot i - \delta \) using \( R \cdot \delta \)
      by (simp-all add: A-def B1-def B3-def exp-eq-polar B2-def Complex-eq
        arc-part-circlepath)

    hence [simp, intro]: valid-path \( \Gamma \)
      by (simp add: \( \Gamma \)-def A-def B1-def B3-def B2-def Complex-eq)
    hence [simp, intro]: path \( \Gamma \) using valid-path-imp-path by blast
    have [simp]: pathfinish \( \Gamma = pathstart \Gamma \) by (simp add: \( \Gamma \)-def exp-eq-polar)

    have image-\( B2 \): path-image \( B2 = \{ s. \ Re s = -\delta \wedge Im s \in \{-R..R\} \} \)
      using \( R \) by (auto simp: closed-segment-same-Re closed-segment-eq-real-ivl
        B2-def)
    have image-\( B1 \): path-image \( B1 = \{ s. \ Re s \in \{-\delta..0\} \wedge Im s = 0 \} \)
      and image-\( B3 \): path-image \( B3 = \{ s. \ Re s \in \{-\delta..0\} \wedge Im s = R \} \)
    using \( \delta \) by (auto simp: B1-def B3-def closed-segment-same-Im closed-segment-eq-real-ivl)
have image-A: path-image $A = \{s. \text{Re } s \geq 0 \wedge \text{norm } s = R\}$
unfolding $A$-def using $R$ by (subst path-image-semicolon-circle-Reg-eq)
also have $z \in \ldots \longrightarrow z \in X' - \{0\}$ for $z$
using complex-Re-le-cmod[of $z$] abs-Im-le-cmod[of $z$] $\delta$ $R$
by (auto simp: X'-def in-cbox-complex-iff)
hence $\{s. \text{Re } s \geq 0 \wedge \text{norm } s = R\} \subseteq X' - \{0\}$ by auto
finally have path-image $B_2 \subseteq X' - \{0\}$ path-image $A \subseteq X' - \{0\}$
by (auto simp: X'-def in-cbox-complex-iff image-B2 image-B1 image-B3)

— $\Gamma$ 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 $\Gamma$ unfolding $\Gamma$-def
proof (intro simple-path-join-loop subsetI arc-join, goal-cases)
fix $z$ assume $z: z \in \text{path-image } A \cap \text{path-image } (B_1 +++ B_2 +++ B_3)$
with image-$A$ have $\text{Re } z \geq 0$ norm $z = R$ by auto
with $z R \delta$ show $z \in \{\text{pathstart } A, \text{pathstart } (B_1 +++ B_2 +++ B_3)\}$
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 $g = (\lambda z::\text{complex}. f (w + z) * N \text{ powr } z * (1 / z + z / R^2))$
define $S$ where $S = \text{eval-fds } (\text{fds-truncate } N F)$
define $g-S$ where $g-S = (\lambda z::\text{complex}. S (w + z) * N \text{ powr } z * (1 / z + z / R^2))$
define $g$-rem where $g$-rem = $(\lambda z::\text{complex}. \text{rem } (w + z) * N \text{ powr } z * (1 / z + z / R^2))$

have $g$-holo: $g$ holomorphic-on $X - \{0\}$ unfolding $g$-def
by (auto intro!: holomorphic-intros analytic-imp-holomorphic[of analytic])

have rem-altdef: $\text{rem } z = \text{eval-fds } (\text{fds-remainder } N F) z$ if $\text{Re } z > 1$ for $z$
proof —
have abscissa: $\text{abs-cone-abscissa } F \leq 1$
using assms by (intro bounded-coeffs-imp-abs-cone-abscissa-le-1)
(simp-all add: natfun-bigo-iff-Bseq)
from assms and that have $f z = \text{eval-fds } F z$ by auto
also have $F = \text{fds-truncate } N F + \text{fds-remainder } N F$
by (rule fds-truncate-plus-remainder [symmetric])
also from that have $\text{eval-fds } \ldots z = S z + \text{eval-fds } (\text{fds-remainder } N F) z$
unfolding $S$-def
by (subst eval-fds-add) (auto intro!: fds-conv-forcing-abs-converges
   fds-conv-forcing[of $\text{le-less-trans}[OF abscissa]]$)

44
finally show \( \text{thesis} \) by (simp add: rem-def)

qed

— We now come to the first application of the residue theorem along the path \( \Gamma \):

\[
\begin{align*}
\text{have } & f \{ \Gamma \} g = 2 \ast \pi \ast i \ast \text{winding-number } \Gamma \ast 0 \ast \text{residue } g \ast 0 \\
\text{proof (subst Residue-theorem)} & \\
\text{show } & g \text{ holomorphic-on } X \ast \{ 0 \} \text{ by fact} \\
\text{show } & \text{path-image } \Gamma \subseteq X \ast \{ 0 \} \text{ using path-images} \\
& \text{by (auto simp: } \Gamma \text{-def path-image-join) \\
\text{thus } & \forall z. z \notin X \longrightarrow \text{winding-number } \Gamma \ast z = 0 \\
& \text{by (auto intro: simply-connected-imp-winding-number-zero[of } X) } \\
& \text{convex-imp-simply-connected} \\
\text{qed (insert path-images, auto intro: convex-connected) } \\
\text{also have } & \text{winding-number } \Gamma \ast 0 = 1 \\
\text{proof (rule simple-closed-path-winding-number-pos) } & \\
\text{from } & R \delta \text{ have } \forall g \in \{ A, B1, B2, B3 \}. \text{Re (winding-number } g \ast 0) > 0 \\
\text{unfolding } & A\text{-def B1-def B2-def B3-def} \\
& \text{by (auto intro!: winding-number-linepath-pos-lt winding-number-part-circlepath-pos-less) } \\
& \text{hence valid-path } \Gamma \ast 0 \notin \text{path-image } \Gamma \ast \text{Re (winding-number } \Gamma \ast 0) > 0 \\
\text{unfolding } & \Gamma\text{-def using path-images(1-4) by (intro winding-number-join-pos-combined')} \\
& \text{auto} \\
& \text{thus } \text{Re (winding-number } \Gamma \ast 0) > 0 \text{ by simp} \\
\text{qed (insert path-images (simple-path } \Gamma\), auto simp: } \Gamma\text{-def path-image-join) } \\
\text{also have } & \text{winding-number } \Gamma \ast 0 = 1 \\
\text{proof (rule simple-closed-path-winding-number-pos) } & \\
\text{from } & R \delta \text{ have } \forall g \in \{ A, B1, B2, B3 \}. \text{Re (winding-number } g \ast 0) > 0 \\
\text{unfolding } & A\text{-def B1-def B2-def B3-def} \\
& \text{by (auto intro!: winding-number-linepath-pos-lt winding-number-part-circlepath-pos-less) } \\
& \text{hence valid-path } \Gamma \ast 0 \notin \text{path-image } \Gamma \ast \text{Re (winding-number } \Gamma \ast 0) > 0 \\
\text{unfolding } & \Gamma\text{-def using path-images(1-4) by (intro winding-number-join-pos-combined')} \\
& \text{auto} \\
& \text{thus } \text{Re (winding-number } \Gamma \ast 0) > 0 \text{ by simp} \\
\text{qed (insert path-images (simple-path } \Gamma\), auto simp: } \Gamma\text{-def path-image-join) } \\
\text{also have } & \text{winding-number } \Gamma \ast 0 = 1 \\
\text{proof (rule simple-closed-path-winding-number-pos) } & \\
\text{from } & R \delta \text{ have } \forall g \in \{ A, B1, B2, B3 \}. \text{Re (winding-number } g \ast 0) > 0 \\
\text{unfolding } & A\text{-def B1-def B2-def B3-def} \\
& \text{by (auto intro!: winding-number-linepath-pos-lt winding-number-part-circlepath-pos-less) } \\
& \text{hence valid-path } \Gamma \ast 0 \notin \text{path-image } \Gamma \ast \text{Re (winding-number } \Gamma \ast 0) > 0 \\
\text{unfolding } & \Gamma\text{-def using path-images(1-4) by (intro winding-number-join-pos-combined')} \\
& \text{auto} \\
& \text{thus } \text{Re (winding-number } \Gamma \ast 0) > 0 \text{ by simp} \\
\text{qed (insert path-images (simple-path } \Gamma\), auto simp: } \Gamma\text{-def path-image-join) } \\
\text{also have } & \text{residue } g \ast 0 = f \ast w \\
\text{proof – } & \\
& \text{have } g = (\lambda z::\text{complex} . f (w + z) \ast N \text{ powr } z \ast (1 + z^2 / R^2) / z) \\
& \text{by (auto simp: } g\text{-def divide-simps fun-eq_iff power2-eq-square \text{ simp del: div-mult-self3 div-mult-self4 div-mult-self2 div-mult-self1)} \text{ by (auto intro!: holomorphic-intros analytic-imp-holomorphic[of } X) } \\
& \text{auto intro!: contour-integral-join contour-integrable-holomorphic-simple } \\
\text{ultimately show } \text{thesis by (simp only:)} \\
\text{qed} \\
\text{finally have } & 2 \ast \pi \ast i \ast f \ast w = f \{ \Gamma \} g \text{ by simp} \\
\text{also have } & \ldots = f \{ A \} g + f \{ B2 \} g + f \{ B1 \} g + f \{ B3 \} g \text{ unfolding } \Gamma\text{-def} \\
& \text{by (auto intro!: contour-integral-join, (insert path-images, auto intro!: contour-integral-join contour-integrable-holomorphic-simple g-holo)[4]+} \\
& \text{(simp-all add: add-ac)} \\
& \text{finally have } \text{integral1: } 2 \ast \pi \ast i \ast f \ast w = f \{ A \} g + f \{ B2 \} g + f \{ B1 \} g + f \{ B3 \} g. \\
\text{— Next, we apply the residue theorem along a circle of radius } R \text{ to another integrand that is related to the partial sum:} \\
\text{have } f \{ \text{circlepath } 0 \ast R \} g-S = 2 \ast \pi \ast i \ast \text{residue } g-S \ast 0 \\
\text{proof (subst Residue-theorem)} & \\
\text{show } g-S \text{ holomorphic-on } \text{UNIV} \ast \{ 0 \} \\
& \text{by (auto simp: g-S-def S-def intro!: holomorphic-intros) }
\end{align*}
\]
\textbf{qed} (insert \(R\), auto simp: winding-number-circlepath-centre)

\textbf{also have} residue \(g-S 0 = S w\)

\textbf{proof} \\
\textbf{have} \(g-S = (\lambda z :: \text{complex}. S (w + z) * N \text{ powr} z * (1 + z^2 / R^2) / z)\)
\begin{itemize}
  \item \textbf{by} (auto simp: \(g-S\)-def divide-simps \(\text{fun-eq-iff}\) power2-eq-square
  \hspace{1em} simp del: \text{div-mul-self}\text{3} \text{div-mul-self}\text{4} \text{div-mul-self}\text{2} \text{div-mul-self}\text{1})
\end{itemize}

\textbf{moreover from} \(N\) \textbf{have} residue \(\ldots 0 = S w\)
\begin{itemize}
  \item \textbf{by} (subst residue-simple[of \(X\)])
  \hspace{1em} (auto intro!: holomorphic-intros simp: \(S\)-def)
\end{itemize}

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

\textbf{qed}

\textbf{finally have} \(2 \ast pi \ast i \ast S w = \oint \text{circlepath 0} R\) \(g-S\) ..

— We split this integral into integrals along two semicircles in the left and right half-plane, respectively:

\textbf{also have} \(\ldots = \oint \text{part-circlepath 0} R \ (\pi / 2) \ (3\pi / 2)\) \(g-S\)

\textbf{proof} (rule Cauchy-theorem-homotopic-loops)

\textbf{show} homotopic-loops \(\{-0\}\) (circlepath 0 \(R\))
\begin{itemize}
  \item \text{part-circlepath 0} \(R \ (-\pi / 2) \ (3\pi / 2)\) \text{unfolding} circlepath-def
\item \textbf{using} \(R\)
\end{itemize}

\begin{itemize}
  \item \textbf{by} (intro homotopic-loops-part-circlepath\{where \(k = 1\)\}) auto
\item \textbf{qed} (auto simp: \(g-S\)-def \(S\)-def intro!: holomorphic-intros)
\end{itemize}

\textbf{also have} \(\ldots = \oint [A \ pluses -A] g-S\)

\textbf{proof} (rule Cauchy-theorem-homotopic-paths)
\begin{itemize}
  \item \textbf{have} \(\ast -A = \text{part-circlepath 0} R \ (\pi / 2) \ (3\pi / 2)\) \text{unfolding} \(A\)-def
  \hspace{1em} (\text{part-circlepath-mirror\{where \(k = 0\)\}}) \text{auto}
\item \textbf{from} \(R\) \textbf{show} \(\text{homotopic-paths} \ (-\{0\})\) \(\text{part-circlepath 0} R \ (-\pi / 2) \ (3\pi / 2)\)
  \hspace{1em} \(\{A \ pluses -A\}\)
\item \textbf{unfolding} \(\ast\) \text{unfolding} \(A\)-def
\end{itemize}

\begin{itemize}
  \item \textbf{by} (intro homotopic-paths-part-circlepath) (auto dest!: \text{in-path-image-part-circlepath})
  \hspace{1em} \textbf{qed} (auto simp: \(g-S\)-def \(A\)-def \text{exp-eq-polar} intro!: holomorphic-intros)
\item \textbf{also have} \(\ldots = \oint [A] g-S + \oint [-A] g-S\) \text{using} \(R\)
\end{itemize}

\begin{itemize}
  \item \textbf{by} (intro \text{contour-integral-join} \text{contour-integrable-holomorphic-simple[of -\{-0\}\}])
  \hspace{1em} (auto simp: \(A\)-def \(g-S\)-def \text{path-image-mirror dest!: in-path-image-part-circlepath}
  \hspace{1em} \text{intro!: holomorphic-intros})
\item \textbf{also have} \(\oint [-A] g-S = -\oint [A] (\lambda x. g-S (-x))\)
\end{itemize}

\begin{itemize}
  \item \textbf{by} (simp add: \(A\)-def \text{contour-integral-mirror} \text{contour-integral-neg})
\end{itemize}

\textbf{finally have} \(\text{integral2}: 2 \ast pi \ast i \ast S w = \oint [A] g-S - \oint [A] (\lambda x. g-S (-x))\)

\item \textbf{by} simp

— Next, we show a small bounding lemma that we will need for the final estimate:

\textbf{have} \(\text{circle-bound}: \text{norm} (1 / z + z / R^2) \leq 2 / R\) if \(\text{simp}: \text{norm} z = R\)

\item \textbf{for} \(z :: \text{complex}\)

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

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.

\begin{equation}
\text{have B12-bound: norm } \left( \int \left( -\delta..0 \right) (\lambda x. g (x + \Re z \cdot i)) \right) \leq 3 * M / R / \ln N
\end{equation}

(is \( \exists l \leq -\) if \( |R| = R \) for \( R' \)

proof —

\begin{enumerate}
\item have \( \exists l \leq \int \left( -\delta..0 \right) (\lambda x. 3 * M / \Re N \powr x) \)
\item proof (rule integral-norm-bound-integral)
\item fix \( z \) assume \( x : x \in \{-\delta..0\} \)
\item define \( z \) where \( z = x + i * R' \)
\item from \( R \) that have \( \exists x : z \neq 0 \Re x z = x \Im x z = R' \)
\item from \( X \) that have \( \norm x \leq \delta + R \)
\item by (intro order.trans[OF cmod_le add-mono]) auto
\end{enumerate}

\begin{enumerate}
\item hence \( \norm \left( 1 / z + z / R^2 \right) \leq 1 / R + (\delta / (R + 1)) / R \)
\item using \( R \) that abs-Im-le-cmod[of x]
\item by (intro order.trans[OF norm-triangle-ineq add-mono])
\item also have \( \delta / R \leq 1 \) using \( \delta R \) by auto
\item finally have \( \norm \left( 1 / z + z / R^2 \right) \leq 3 / R \)
\item using \( R \) by (simp add: divide-right-mono)
\item hence \( \norm \left( g z \right) \leq M * N \powr x * (3 / R) \)
\item unfolding g-def norm-mul using \( M \geq 0 \), \( \langle z \in X' \rangle \)
\item by (intro mult-mono mult-nonneg-nonneg M) (auto simp: norm-powr-real-powr)
\item thus \( \norm \left( g (x + R' * i) \right) \leq 3 * M / \Re N \powr x \) by (simp add: mult-ac z-def)
\item qed (insert \( N R l \) that \( \delta \), auto intro!: integrable-continuous-real-continuous-intros)
\item simp: g-def X-def complex-eq-iff in-box-complex-iff)
\item also have \( \ldots = 3 * M / \Re N * \int \left( -\delta..0 \right) (\lambda x. N \powr x) \) by simp
\item also have \( \langle \lambda x. N \powr x \rangle \) has-integral \( \left( N \powr 0 / \ln N - N \powr (-\delta) / \ln N \right) \) \(-\delta..0\)
\item using \( \delta \)
\item by (intro fundamental-theorem-of-calculus)
\item (auto simp: has-real-derivative-iff-has-vector-derivative [symmetric]
\item powr-def
\item intro!: derivative-eq-intros)
\item hence \( \int \left( -\delta..0 \right) (\lambda x. N \powr x) = 1 / \ln (\real N) - \real N \powr -\delta / \ln (\real N)
\item using \( N \) by (simp add: has-integral-iff)
\item also have \( \ldots \leq 1 / \ln (\real N) \) using \( N \) by simp
\item finally show \( \text{?thesis using } M R \) by (simp add: mult-left-mono divide-right-mono)
\item qed

\end{enumerate}
We combine the two results from the residue theorem and obtain an integral representation of the difference between the partial sums and the limit:

\[
\begin{aligned}
&\text{have } 2 * \pi * i * (f w - S w) = \\
&\quad \int [A] g - \int [A] g - S + \int [A] (\lambda x. g - S (-x)) + \int [B1] g + \int [B3] g + \\
&\quad \int [B2] g .
\end{aligned}
\]

unfolding ring-distributes integral1 integral2 by (simp add: algebra-simps)
also have \(\int [A] g - \int [A] g - S = \int [A] (\lambda x. g x - g - S x)\) using path-images
by (intro contour-integral-diff [symmetric])
(auto intro: contour-integrable-holomorphic-simple[of - X - {0}] holomorphic-intros
simp: g-S-def g-holo S-def)
also have \(\ldots = \int [A] g - \text{rem}\)
by (simp add: g-rem-def g-S-def algebra-simps rem-def)
finally have \(2 * \pi * i * (f w - S w) = \\
\quad \int [A] g - \text{rem} + \int [A] (\lambda x. g - S (-x)) + \int [B1] g + \int [B3] g + \\
\quad \int [B2] g .
\]

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

\[
\begin{aligned}
&\text{auto} \\
&\text{finally show } \text{norm} (\int [B3] g) \leq \ldots .
\end{aligned}
\]
next
have \(\text{norm} (\int [B2] g) \leq M * N \text{ powr} (-\delta) * (3 / \delta) * \\
\text{norm} (\text{Complex} (-\delta) (-R) - \text{Complex} (-\delta) R)\) unfolding B2-def
proof ((rule contour-integral-bound-linepath; (fold B2-def)?) goal-cases)
\(\text{case} \(3 z\)\)
from \(3 \delta R\) have \([\text{simp}]: z \neq 0\) and \(\text{Re}\)-\(z\): \(\text{Re} \ z = -\delta\) and \(\text{Im}\)-\(z\): \(\text{Im} \ z \in \{-R..R\}\)
by (auto simp: closed-segment-same-Re closed-segment-eq-real-interval)
from \(3 \delta R\) have \(\text{norm} z \leq \sqrt{\delta^2 + R^2}\) unfolding cmod-def using \(\text{Re}\)-\(z\) \(\text{Im}\)-\(z\)
by (intro real-sqrt-le_mono add-mono) (auto simp: power2-le-iff-abs-le)
from \(\text{power-mono}\) \([\text{OF this}, of 2]\) have \(\text{norm-sqr} \ (\text{norm} z \cdot 2 \leq \delta^2 + R^2)\)
by simp

have norm (1 / z + z / R^2) ≤ (1 + (norm z)^2 / R^2) / δ
unfolding add-divide-distrib using δ R abs-Re-le-cmod[of z]
by (intro order.trans[OF norm-triangle-ineq] add-mono)
(auto simp: norm-dive norm-mult field-simps power2-eq-square Re-z)
also have "... ≤ (1 + (1 + δ^2 / R^2)) / δ using δ R z ∈ 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 δ^2 / R^2 ≤ 1
using δ R by (auto simp: field-simps intro!: power-mono)
finally have norm (1 / z + z / R^2) ≤ 3 / δ using δ by(simp add: divide-right-mono)
with z ∈ X' show norm (g z) ≤ M * N powr (- δ) * (3 / δ) unfolding g-def norm-mult
by (intro mult-mono mult-nonneg-nonneg M) (auto simp: norm-power-real-powr Re-z)
qed (insert path-images M δ, auto intro!: contour-integrable-holomorphic-simple[OF g-holo])
thus norm (\| [A] g) ≤ 6 * R * M * N powr (- δ) / δ
using R by (simp add: field-simps cmod-def real-sqrt-mult)
next
have norm (\{ [A] (λx. g-S (- x))) ≤ (2 * C / (real N * R) + 2 * C / R^2)
* R * ((π/2) − (−π/2)) unfolding A-def
proof (rule contour-integrable-bound-part-circlepath-strong[where k = {R * i, −R * i}];
(fold A-def)?)
by auto

have norm (g-S (- z)) ≤ ...
by (simp add: S-def eval-fds-truncate)
also have norm ... ≤ C * N powr Re z * (1 / N + 1 / Re z)
using Re z > 0, w N by (intro newman-ingham-aux2 C)
finally have norm (S (w − z)) ≤ ...

hence norm (g-S (- z)) ≤ C * N powr (Re z) * (1 / N + 1 / Re z) * N powr (−Re z) * (2
* Re z / R^2)
unfolding g-S-def norm-mult
using newman-ingham-aux1[OF - (norm z = R) (Re z > 0) (C ≥ 1) R]
by (intro mult-mono mult-nonneg-nonneg circle-bound)
(auto simp: norm-power-real-powr norm-uminus-minus)
also have ... = 2 * C * (Re z / (N + 1)) / R^2 using R N (Re z > 0)
by (simp add: powr-minus algebra-simps)
also have \( \ldots \leq 2 \cdot C / (N \cdot R) + 2 \cdot C / R^2 \) unfolding add-divide-distrib
ring-distrib
using \( R \cdot N \cdot \text{abs-}\text{Re-}\text{-}\text{le-cmod[of z]} \langle \text{norm z} = R \cdot (\text{Re z} > 0) \rangle \langle C \geq 1 \rangle \)
by (intro add-mono) (auto simp: power2-eq-square field-simps mult-mono)
finally show \(?\)case .
qed (insert \( R \cdot N \cdot \text{image-A C} \), auto intro: \text{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 \( R \cdot N \)
by (simp add: power2-eq-square field-simps)
finally show \( \text{norm} (\langle f [A] (\lambda x. g \cdot S (- z)) \rangle) \leq \ldots \).
next
have \( \text{norm} (\langle f [A] g \cdot \text{rem} \rangle) \leq (2 \cdot C / R^2) \cdot R \cdot ((\pi / 2) - (-\pi / 2)) \)
unfolding A-def
proof ((rule contour-integrable-bound-part-circlepath-strong)[where \( k = \{ R \cdot i, -R \cdot i \} \])
(fold A-def)\?) goal-cases
\( \langle \text{case} (6 z) \rangle \) 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 \( \langle \text{norm z} = R \rangle \) 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 \( \text{summable: summable} (\lambda n. C \cdot (1 / (\text{Suc n} + N)) \cdot \text{powr} (\text{Re w + Re z})) \)
using \( \text{summable-hurwitz-zeta-real[of Re w + Re z Succ N]} \langle \text{Re z} > 0 \rangle \cdot w \)
unfolding powr-minus by (intro summable-mult) (auto simp: field-simps)
have \( \text{rem} \ (w + z) = (\sum n. \text{fds-nth F \ (Suc n + N) / (Suc n + N) \ powr (w + z)}) \)
using \( \langle \text{Re z} > 0 \rangle \cdot w \) by (simp add: rem-altdef eval-fds-remainder)
also have \( \text{norm} \ldots \leq (\sum n. C / (\text{Suc n} + N) \cdot \text{powr} \ (\text{Re w + Re z})) \) using
summable
by (intro norm-suminf-le)
(auto simp: norm-divide norm-powr-powr intro!: divide-right-mono C)
also have \( \ldots = (\sum n. C \cdot (\text{Suc n} + N) \cdot \text{powr} \ (\text{Re w + Re z})) \)
unfolding powr-minus by (simp add: field-simps)
also have \( \ldots = C \cdot (\sum n. (\text{Suc n} + N) \cdot \text{powr} \ (\text{Re w + Re z})) \)
using \( \text{summable-hurwitz-zeta-real[of Re w + Re z Succ N]} \langle \text{Re z} > 0 \rangle \cdot w \)
by (subst suminf-mult) (auto simp: add-ac)
also have \( (\sum n. (\text{Suc n} + N) \cdot \text{powr} \ (\text{Re w + Re z})) \leq N \cdot \text{powr} (1 - (\text{Re w + z})) / (\text{Re w + z} - 1) \)
using \( \langle \text{Re z} > 0 \rangle \cdot w \cdot N \cdot \text{hurwitz-zeta-real-bound-aux[of N Re w + z]} \)
by (auto simp: add-ac)
also have \( \ldots \leq N \cdot \text{powr} \ (-\text{Re z} / \text{Re z}) \)
using \( w \cdot N \cdot \langle \text{Re z} > 0 \rangle \) by (intro frac-le powr-mono) auto

50
finally have \( \norm{(\rem (w + z))} \leq C / (\Re z \ast N \powr \Re z) \)
using \( C \) by (simp add: mult-left-mono mult-right-mono powr-minus field-simps)
hence \( \norm{(g \cdot \rem z)} \leq (C / (\Re z \ast N \powr \Re z)) \ast N \powr (\Re z) \ast (2 \ast \Re z / R^2) \)
unfolding \( g \cdot \rem-def \) norm-mult
using \( \text{newman-ingham-aux1} [\OF - \langle \norm z = R \rangle \ast R \langle \Re z > 0 \rangle \ast C \]
by (intro mult-mono mult-nonneg-nonneg circle-bound)
(auto simp: norm-powr-real-powr norm-uminus-minus)
also have \( \ldots = 2 \ast C / R^2 \) using \( R \ast N \langle \Re z > 0 \rangle \)
by (simp add: powr-minus field-simps)
finally show \( \langle \case \rangle . \)
next
show \( g \cdot \rem \text{ contour-integrable-on } A \) using \( \text{path-images} \)
by (auto simp: g-rem-def rem-def S-def
    intro!: contour-integrable-holomorphic-simple[af - X-{0}])
holomorphic-intros)
qed (insert \( R \ast N \ast C \), auto)
also have \( \langle 2 \ast C / R^2 \rangle \ast R \ast ((\pi/2) - (-\pi/2)) = 2 \ast C \ast \pi / R \)
using \( R \) by (simp add: power2-eq-square field-simps)
finally show \( \langle \norm{(f \ast [A] \ g-rem)} \rangle \leq \ldots \).
\\( \text{qed} \)
also have \( \ldots = 4 \ast C \ast \pi / R + 2 \ast C \ast \pi / N + 6 \ast M / R \ast \ln N + 6 \ast R \ast M \ast N \)
\\( \powr - \delta / \delta \)
by (simp add: algebra-simps)
also have \( \ldots = 2 \ast \pi \ast (2 \ast C / R + C / N + 3 \ast M / (\pi \ast R \ast \ln N) + 3 \ast R \ast M \)
\\( / (\delta \ast \pi \ast N \powr \delta) \))
by (simp add: field-simps powr-minus)
also have \( \langle \norm{(2 \ast \pi \ast i \ast (\f w - S w))} = 2 \ast \pi \ast \norm{(\f w - S w)} \rangle \)
by (simp add: norm-mult)
finally have \( \langle \norm{(\f w - S w)} \rangle \leq \langle \text{bound} \rangle \) by (simp add: bound-def)
also have \( \langle \text{bound} \rangle < \varepsilon \) by fact
finally show \( \langle \f w - S w \rangle < \varepsilon \).
\\( \text{qed} \)
\\( \text{qed} \)
thus \( \text{fds-converges} \ f \ w \)
by (auto simp: fds-converges-altdef2 intro: convergentI)
thus \( \text{eval-fds} \ F \ f \ w = f \ w \)
using \( \langle \langle \Lambda N. \ \text{eval-fds} \ (\text{fds-truncate} \ N \ F) \ w \rangle \longrightarrow f \ w \rangle \)
by (intro tendsto-unique[\OF - tendsto-\text{eval-fds-truncate}]) auto
\\( \text{qed} \)

The theorem generalises in a trivial way; we can replace the requirement that the coefficients of \( f(s) \) be \( O(1) \) by \( O(n^{\sigma-1}) \) for some \( \sigma \in \mathbb{R} \), then \( f(s) \) converges for \( \Re(s) > \sigma \). If it can be analytically continued to \( \Re(s) \geq \sigma \), it is also convergent there.

\textbf{Theorem} Newman-Ingham:

\textbf{Fixes} \( F :: \text{complex} \) \textbf{and} \( f :: \text{complex} \Rightarrow \text{complex} \)
\textbf{Assumes} \( \text{coeff-bound} :: \text{fds-nth} \ F \in O(\lambda n. \ n \powr \text{of-real} \ (\sigma - 1)) \)

51
assumes f-analytic: \( f \text{ analytic-on } \{ s. \ Re s \geq \sigma \} \)
assumes F-conv-f: \( \forall s. \ Re s > \sigma \implies \text{eval-fds } F s = f s \)
assumes w: \( \Re w \geq \sigma \)
shows fds-converges \( F w \) and eval-fds \( F w = f w \)
proof –
define \( F' \) where \( F' = \text{fds-shift} (-\text{of-real} (\sigma - 1)) F \)
define \( f' \) where \( f' = f \circ (\lambda s. s + \text{of-real} (\sigma - 1)) \)

have fds-nth \( F' = (\lambda n. \text{fds-nth } F n * \text{of-nat } n \text{powr} -\text{of-real}(\sigma - 1)) F \)
  by (auto simp: fun-eq-iff \( F' \)-def)
also have ... \( \in O(\lambda n. \text{of-nat } n \text{powr} -\text{of-real}(\sigma - 1)) \)
  by (intro landau-o.big.mult-right assms)
also have \( (\lambda n. \text{of-nat } n \text{powr} -\text{of-real}(\sigma - 1)) \in \Theta(\lambda --. 1) \)
  by (intro bigthetaI-cong eventually-mono[OF eventually-gt-at-top[of 0]])
(auto simp: powr-minus powr-diff)
finally have \( \text{bigo: } \text{fds-nth } F' \in O(\lambda --. 1). \)

from f-analytic have analytic: \( f' \text{ analytic-on } \{ s. \ Re s \geq 1 \} \) unfolding \( f' \)-def
  by (intro analytic-on-compose-gen[OF - f-analytic]) (auto intro analytic-intros)

have \( F'f: \text{eval-fds } F' s = f' s \) if \( \Re s > 1 \) for \( s \)
  using assms that by (auto simp: \( F' \)-def \( f' \)-def algebra-simps)

have \( w': 1 \leq \Re (w - \text{of-real} (\sigma - 1)) \)
  using \( w \) by simp

have 1: fds-converges \( F' (w - \text{of-real} (\sigma - 1)) \)
  using \( \text{bigo analytic } F'f w' \) by (rule Newman-Ingham-1)
thus fds-converges \( F w \) by (auto simp: \( F' \)-def)

have 2: eval-fds \( F' (w - \text{of-real} (\sigma - 1)) = f' (w - \text{of-real} (\sigma - 1)) \)
  using \( \text{bigo analytic } F'f w' \) by (rule Newman-Ingham-1)
thus eval-fds \( F w = f w \)
  using assms by (simp add: \( F' \)-def \( f' \)-def)
qed

end

3 Prime-Counting Functions
	heory Prime-Counting-Functions
  imports Prime-Number-Theorem-Library
begin

We will now define the basic prime-counting functions \( \pi, \vartheta, \text{ and } \psi \). Additionally, we shall define a function \( M \) that is related to Mertens’ theorems and Newman’s proof of the Prime Number Theorem. Most of the results in
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 p. 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 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)
  by (subst sum-list-distinct-cone-sum-set (auto simp: set-primes-upto conj-commute))
thus ?thesis by (simp add: prime-sum-upto-altdef2)

**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)

**Lemma** prime-sum-upto-eqI'
assumes a' ≤ nat ⌊a⌋ a ≤ b nat ⌊b⌋ ≤ b' ⋀ 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 (λ p. of-nat (f p)) x
by (simp add: prime-sum-upto-def)
lemma prime-sum-upto-mono:
  assumes \( \forall n. n > 0 \implies f n \geq (0::\text{real}) \) \( x \leq y \)
  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 \( \forall n. n > 0 \implies f n \geq (0::\text{real}) \)
  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 \( \lfloor x \rfloor = 0 \lor \lfloor x \rfloor = 1 \) 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 [measurable]: \( \forall y. (\lambda t. f t y) \in M \rightarrow M \) borel
  assumes [measurable]: \( x \in M \rightarrow M \) 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 \( \forall n. \neg \text{primepow} n \implies f n = 0 \)
  \( \forall p i. \text{prime} p \implies i > 0 \implies f (p ^ i) = g p i \)
  shows \( \text{sum-upto} f x = (\sum (p, i) \mid \text{prime} p \land i > 0 \land \text{real} (p ^ i) \leq x). g p i \)
  proof -
    let \( ?d = \text{aprimedivisor} \)
    have \( g : g (?d n) \) (multiplicity (?d n) n) = \( f n \) if primepow n for n using that
      by (subst assms[2] [symmetric])
    (auto simp: primepow-decompose aprime divisor-prime power primepow gt Suc 0
          intro: aprime divisor nat multiplicity aprime divisor gt 0 nat)
    have \( \text{sum-upto} f x = (\sum n \mid \text{primepow} n \land \text{real} n \leq x. f n) \)
    unfolding sum-upto def using assms
    by (intro sum mono neutral cong right) (auto simp: primepow gt 0 nat)
    also have \( \ldots \) \( = (\sum (p, i) \mid \text{prime} p \land i > 0 \land \text{real} (p ^ i) \leq x. g p i) \) (is - = sum - ?S)
      by (rule sum reindex bij witness[of - \lambda (p,i). p ^ i \lambda n. (?d n, multiplicity (?d n) n)])
    (auto simp: aprime divisor prime power primepow decompose primepow gt Suc 0
           ...)
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 (M)

end

**bundle no-prime-counting-notation**

begin

no-notation primes-pi (π)
no-notation primes-theta (ϑ)
no-notation primes-psi (ψ)
no-notation primes-M (M)

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-M = 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 → 𝒜 borel
and measurable-ϑ [measurable]: ϑ ∈ borel → 𝒜 borel
and measurable-ψ [measurable]: ψ ∈ borel → 𝒜 borel
and measurable-primes-M [measurable]: 𝒜 ∈ borel → 𝒜 borel

unfolding primes-M-def π-def ϑ-def ψ-def by measurable
lemma π-eq-0 [simp]: \( x < 2 \implies \pi x = 0 \)
and θ-eq-0 [simp]: \( x < 2 \implies \theta x = 0 \)
and primes-M-eq-0 [simp]: \( x < 2 \implies M x = 0 \)

unfolding primes-pi-def primes-theta-def primes-M-def

by (rule prime-sum-upto-eq-0; simp)+

lemma π-nat-cancel [simp]: \( \pi (nat \ x) = \pi x \)
and ϑ-nat-cancel [simp]: \( \theta (nat \ x) = \theta x \)
and primes-M-nat-cancel [simp]: \( M (nat \ x) = M x \)
and π-floor-cancel [simp]: \( \pi (of-int \ \lfloor y \rfloor) = \pi y \)
and ϑ-floor-cancel [simp]: \( \theta (of-int \ \lfloor y \rfloor) = \theta y \)
and primes-M-floor-cancel [simp]: \( M (of-int \ \lfloor y \rfloor) = M y \)
and ψ-floor-cancel [simp]: \( \psi (of-int \ \lfloor y \rfloor) = \psi y \)

by (simp-all add: π-def θ-def ψ-def primes-M-def prime-sum-upto-altdef2 sum-upto-altdef)

lemma π-nonneg [intro]: \( \pi x \geq 0 \)
and θ-nonneg [intro]: \( \theta x \geq 0 \)
and primes-M-nonneg [intro]: \( M x \geq 0 \)

unfolding primes-pi-def primes-theta-def primes-M-def

by (rule prime-sum-upto-nonneg; simp)+

lemma π-mono [intro]: \( x \leq y \implies \pi x \leq \pi y \)
and θ-mono [intro]: \( x \leq y \implies \theta x \leq \theta y \)
and primes-M-mono [intro]: \( x \leq y \implies M x \leq M y \)

unfolding primes-pi-def primes-theta-def primes-M-def

by (rule prime-sum-upto-mono; simp)+

lemma π-pos-iff: \( \pi x > 0 \longleftrightarrow x \geq 2 \)

proof
  assume x: \( x \geq 2 \)
  show \( \pi x > 0 \)
    by (rule less-le-trans[OF - π-mono[OF x]]) (auto simp: eval-π)

next
  assume π x > 0
  hence \( \neg(x < 2) \) by auto
  thus \( x \geq 2 \) by simp

qed

lemma π-pos: \( x \geq 2 \implies \pi x > 0 \)

by (simp add: π-pos-iff)

lemma ψ-eq-0 [simp]:
  assumes \( x < 2 \)
  shows \( \psi x = 0 \)

proof
  from assms have \( nat \ [x] \leq 1 \) by linarith
  hence mangoldt n = (0 :: real) if \( n \in \{0<..nat \ [x]\} \) 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 {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 0 show ?thesis by (simp add: eval-π)
next
  case (Suc n)
  define n' where n' = smallest-prime-beyond (Suc n)
  have n < n'
     using smallest-prime-beyond-le[of Suc n] unfolding n'-def by linarith
  have prime n' by (simp add: n'-def)
  have n' ≤ p if prime p p > n for p
     using that smallest-prime-beyond-smallest[of p Suc n] by (auto simp: n'-def)
  note n' = ‹n < n'› ‹prime n'› this
  have π (real n') = real (card {p. prime p ∧ p ≤ n'})
     by (subst π-smallest-prime-beyond)
  also have Suc n ≤ n' unfolding n'-def by (rule smallest-prime-beyond-le)
  hence {p. prime p ∧ p ≤ n'} = {p. prime p ∧ p ≤ n} ∪ {p. prime p ∧ p ∈ {n<n'}}
     by auto
  also have real (card ... ) = π (real n) + real (card {p. prime p ∧ p ∈ {n<n'}})
     by (subst card-Un-disjoint) (auto simp: π-def prime-sum-upto-def)
  also have {p. prime p ∧ p ∈ {n<n'}} = {n'}
     using n' by (auto intro: antisym)
  finally show ?thesis using Suc by (simp add: n'-def)
qed

lemma π-inverse-exists: ∃n. π (real n) = real m
proof (induction m)
  case 0
  show ?case by (intro exI[of - 0]) auto
next
  case (Suc m)
  from Suc.IH obtain n where n: π (real n) = real m
     by auto
  hence π (real (smallest-prime-beyond (Suc n))) = real (Suc m)
     by (subst π-smallest-prime-beyond) auto
  thus ?case by blast
qed

lemma nth-prime-exists: ∃p::nat. prime p ∧ card {q. prime q ∧ q < p} = n
proof
  from π-inverse-exists[of n] obtain m where π (real m) = real n by blast
  hence card: card {q. prime q ∧ q ≤ m} = n
     by (auto simp: π-def prime-sum-upto-def)
  define p where p = smallest-prime-beyond (Suc m)
  have m < p using smallest-prime-beyond-le[of Suc m] unfolding p-def by linarith
  have prime p by (simp add: p-def)
  have p ≤ q if prime q q > m for q
     using smallest-prime-beyond-smallest[of q Suc m] that by (simp add: p-def)
note $p = \langle m < p, \prime p \rangle$ this

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

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

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

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

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

lemma nth-prime-eqI'':

59
assumes \( p \) \( \pi (\text{real } p) = \text{real } n + 1 \)
shows \( \text{nth-prime } n = p \)
proof (rule nth-prime-eqI')
  have \( \text{real } (\text{card } \{ q. \text{prime } q \land q \leq p\}) = \pi (\text{real } p) \)
    by (simp add: \( \pi \)-def prime-sum-upto-def)
  also have \( \ldots = \text{real } (\text{Suc } n) \) by (simp add: assms)
  finally show \( \text{card } \{ q. \text{prime } q \land q \leq p\} = \text{Suc } n \)
    by (simp only: of-nat-eq-iff)
qed
lemma \( \text{n-th-prime-at-top: } \limsup_{n \to \infty} \text{n-th-prime} = \infty \)
proof (rule filterlim-at-top-monotone)
show \( \limsup (\lambda n:nat. \; n + 2) = \infty \) by \( \text{real-asym} \)
qed (auto simp: \( \text{n-th-prime-lower-bound} \))

lemma \( \pi-at-top: \limsup \pi = \infty \)
unfolding filterlim-at-top
proof
safe
fix \( C : real \)
define \( x_0 \) where \( x_0 = \text{real} \; (\text{n-th-prime} (\text{nat} \; \lceil \text{max} \; 0 \; C \rceil)) \)
show \( \text{eventually} (\lambda x. \pi x \geq C) \) at-top
using \( \text{eventually-ge-at-top} \)
proof
eventually-elim
fix \( x \)
assume \( x \geq x_0 \)
have \( C \leq \text{real} \; (\text{nat} \; \lceil \text{max} \; 0 \; C \rceil + 1) \) by linarith
also have \( \text{real} \; (\text{nat} \; \lceil \text{max} \; 0 \; C \rceil + 1) = \pi x_0 \)
unfolding \( x_0 \)-def by simp
also have \( \ldots \leq \pi x \) by (rule \( \pi \)-mono) fact
finally show \( \pi x \geq C \).
qed

qed

An unbounded, strictly increasing sequence \( a_n \) partitions \( [a_0; \infty) \) into segments of the form \( [a_n; a_{n+1}) \).

lemma \( \text{strict-mono-sequence-partition:} \)
assumes \( \text{strict-mono} (f :: \text{nat} \Rightarrow 'a :: \{\text{linorder}, \text{no-top}\}) \)
assumes \( x \geq f 0 \)
assumes \( \limsup f = \infty \)
shows \( \exists k. \; x \in [f k..f (Suc k)) \)
proof
\{ define \( k \) where \( k = (\text{LEAST} \; k. \; f (Suc k) > x) \)

obtain \( n \) where \( x \leq f n \)
using assms by (auto simp: filterlim-at-top eventually-at-top-linorder)
also have \( f n < f (Suc n) \)
using assms by (auto simp: strict-mono-Suc-iff)
finally have \( \exists n. \; f (Suc n) > x \) by auto
\}
from LeastI-ex[OF this] have \( x < f (Suc k) \)
by (simp add: \( k \)-def)
moreover have \( f k \leq x \)
proof (cases \( k \))
case \( Suc k' \)
have \( k \leq k' \) if \( f (Suc k') > x \)
using that unfolding \( k \)-def by (rule Least-le)
with \( Suc \) show \( f k \leq x \) by (cases \( f k \leq x \)) (auto simp: not-le)
qed (use assms in auto)
ultimately show \( \vdots \)thesis by auto
qed
lemma nth-prime-partition:
  assumes \( x \geq 2 \)
  shows \( \exists k. \ x \in \{ \text{nth-prime } k..<\text{nth-prime } (\text{Suc } k) \} \)
  using strict-mono-sequence-partition[OF strict-mono-nth-prime, of \( x \)] assms nth-prime-at-top
  by simp

lemma nth-prime-partition':
  assumes \( x \geq 2 \)
  shows \( \exists k. \ x \in \{ \text{real } (\text{nth-prime } k)..<\text{real } (\text{nth-prime } (\text{Suc } k)) \} \)
  by (rule strict-mono-sequence-partition)
    (auto simp: strict-mono-Suc-iff assms intro!: filterlim-real-sequentially filterlim-compose[OF - nth-prime-at-top])

lemma between-nth-primes-imp-nonprime:
  assumes \( n > \text{nth-prime } k \ n < \text{nth-prime } (\text{Suc } k) \)
  shows \( \neg \text{prime } n \)
  using assms
  by (metis Suc-leI not-le nth-prime-Suc smallest-prime-beyond-smallest)

lemma nth-prime-partition'':
  assumes \( x \geq (2 :: \text{real}) \)
  shows \( x \in \{ \text{real } (\text{nth-prime } (\text{nat } \lfloor \pi \ x \rfloor - 1))..<\text{real } (\text{nth-prime } (\text{nat } \lfloor \pi \ x \rfloor)) \} \)
proof -
  obtain \( n \) where \( n: \ x \in \{ \text{nth-prime } n..<\text{nth-prime } (\text{Suc } n) \} \)
    using nth-prime-partition' assms by auto
  have \( \pi (\text{nth-prime } n) = \pi x \)
    unfolding \( \pi - \text{def} \)
  using between-nth-primes-imp-nonprime \( n \)
    by (intro prime-sum-upto-eqI)
    (auto simp: le-nat-iff le-floor-iff)
  hence \( n = \text{nat } \lfloor \pi x \rfloor - 1 \)
    by simp
  hence \( n + 1 = \text{nat } \lfloor \pi x \rfloor \)
    by linarith
  with \( n \)
  show \( \text{thesis} \)
    by simp
qed

3.4 Relations between different prime-counting functions

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

lemma \( \psi - \text{aldef} \):
  assumes \( x > 0 \)
  shows \( \psi x = \text{sum-upto } (\lambda m. \text{prime-sum-upto } \ln (\text{root } m \ x)) (\text{log } 2 \ x) \) (is \( ?rhs \))
proof -
  have finite: finite \( \{ p. \ \text{prime } p \ \land \ \text{real } p \leq y \} \) for \( y \)
    by (rule finite-subset[of \( \{ \text{nat } \lfloor y \rfloor \} \)]
    (auto simp: le-nat-iff le-floor-iff)
  define \( S \) where \( S = (\Sigma i: \{ i. \ \theta < i \ \land \ \text{real } i \leq \text{log } 2 \ x \}) \)
  (\text{p. prime } p \ \land \ \text{real } p \leq \text{root } i \ x \})
  have \( \psi x = (\sum (p, i) | \ \text{prime } p \ \land \ \theta < i \ \land \ \text{real } (p ^ i) \leq x. \ln (\text{real } p)) \)
    unfolding
\[ \psi \text{-def} \]

by (subst sum-upto-primepow[where \( g = \lambda p \cdot \ln (\text{real} \ p) \)])
(auto simp: case-prod-unfold mangoldt-non-primepow)
also have \( \ldots = (\sum (i, p) \mid \text{prime} \ p \land 0 < i \land \text{real} \ p \leq x \cdot \ln \ (\text{real} \ p)) \)
by (intro sum.reindex-bij-witness[of \(-\cdot\lambda(x,y). (y,x)\람(x,y). (y,x))] auto
also have \( \{(i, p). \text{prime} \ p \land 0 < i \land \text{real} \ p \leq x \} = S \)
unfolding S-def
proof safe
fix \( i \cdot p :: \text{nat} \) assume \( i > 0 \) real \( i \leq \log 2 \cdot x \) prime \( p \) real \( p \leq \text{root} \ i \ x \)
hence real \( \langle p \cdot i \rangle \leq \text{root} \ i \ x \cdot i \) unfolding of-nat-power by (intro power-mono)
auto
with \( \langle i, p \rangle \) assms show \( \text{real} \ p \leq \text{root} \ i \ x \)
next
fix \( i \cdot p \) assume \( \langle i, p \rangle \) prime \( p \) \( i > 0 \) real \( p \cdot i \leq x \)
from \( \langle i, p \rangle \) have \( 2 \cdot i \leq p \cdot i \) by (intro power-mono) (auto dest: prime-gt-1-nat)
also have \( \ldots \leq x \) using \( \langle i, p \rangle \) assms
finally show \( \text{real} \ i \leq \log 2 \cdot x \)
using \( \langle i, p \rangle \) assms by (simp add: le-log-iff powr-realpow)
have \( \text{root} \ i \ (\text{real} \ p \cdot i) \leq \text{root} \ i \ x \) using \( \langle i, p \rangle \) assms
by (subst real-root-le-iff) auto
also have \( \text{root} \ i \ (\text{real} \ p \cdot i) = \text{real} \ p \)
using \( \langle i, p \rangle \) assms by (subst real-root-pos2) auto
finally show \( \text{real} \ p \leq \text{root} \ i \ x \).
qed
also have \( (\sum (i,p) \in S. \ln \ p) = \text{sum-upto} \ (\lambda m. \text{prime-sum-upto} \ln \ (\text{root} \ m \ x)) \)
(autosimp: le-log-iff powr-realpow)
also have \( \langle i, p \rangle \in S. \ln \ p = \text{sum-upto} \ (\lambda m. \text{prime-sum-upto} \ln \ (\text{root} \ m \ x)) \)
(auto simp: case-prod-unfold mangoldt-non-primepow)
also have \( \{(i, p). \text{prime} \ p \land 0 < i \land \text{real} \ p \leq x \} = S \)
unfolding S-def
proof safe
fix \( i \cdot p :: \text{nat} \) assume \( i > 0 \) real \( i \leq \log 2 \cdot x \) prime \( p \) real \( p \leq \text{root} \ i \ x \)
hence real \( \langle p \cdot i \rangle \leq \text{root} \ i \ x \cdot i \) unfolding of-nat-power by (intro power-mono)
auto
with \( \langle i, p \rangle \) assms show \( \text{real} \ p \leq \text{root} \ i \ x \)
next
fix \( i \cdot p \) assume \( \langle i, p \rangle \) prime \( p \) \( i > 0 \) real \( p \cdot i \leq x \)
from \( \langle i, p \rangle \) have \( 2 \cdot i \leq p \cdot i \) by (intro power-mono) (auto dest: prime-gt-1-nat)
also have \( \ldots \leq x \) using \( \langle i, p \rangle \) assms
finally show \( \text{real} \ i \leq \log 2 \cdot x \)
using \( \langle i, p \rangle \) assms by (simp add: le-log-iff powr-realpow)
have \( \text{root} \ i \ (\text{real} \ p \cdot i) \leq \text{root} \ i \ x \) using \( \langle i, p \rangle \) assms
by (subst real-root-le-iff) auto
also have \( \text{root} \ i \ (\text{real} \ p \cdot i) = \text{real} \ p \)
using \( \langle i, p \rangle \) assms by (subst real-root-pos2) auto
finally show \( \text{real} \ p \leq \text{root} \ i \ x \).
qed

lemma \( \psi \)-conv-\( \theta \)-sum: \( x > 0 \implies \psi \cdot x = \text{sum-upto} \ (\lambda m. \theta \ (\text{root} \ m \ x)) \)
(autosimp: le-log-iff powr-realpow)
lemma \( \psi \)-minus-\( \theta \):
assumes \( x \cdot x \geq 2 \)
shows \( \psi \cdot x \cdot \theta + x = (\sum i \mid 2 \leq i \land \text{real} \ i \leq \log 2 \cdot x \cdot \theta \ (\text{root} \ i \ x)) \)
proof –
have \( \text{finite} \ \text{finite} \ \{i. \ 2 \leq i \land \text{real} \ i \leq \log 2 \cdot x\} \)
by (rule finite-subset[of \{-2..nat \ [\log 2 \cdot x]\}]) (auto simp: le-log-iff le-floor-iff)
have \( \psi \cdot x \cdot \theta + x = (\sum i \mid 0 < i \land \text{real} \ i \leq \log 2 \cdot x \cdot \theta \ (\text{root} \ i \ x)) \)
using \( x \)
by (simp add: \psi\cdot\theta\cdot\sum \text{sum-upto-def})
also have \( \{i. \ 0 < i \land \text{real} \ i \leq \log 2 \cdot x\} = \text{insert} \ 1 \ \{i. \ 2 \leq i \land \text{real} \ i \leq \log 2 \cdot x\} \)
using \( x \)
by (auto simp: le-log-iff)
also have \( \sum i \mid \theta \ (\text{root} \ i \ x) \cdot \theta = \sum i \mid 2 \leq i \land \text{real} \ i \leq \log 2 \cdot x \cdot \theta \ (\text{root} \ i \ x)) \)
using \( \text{finite} \)
by (subst \text{sum.insert}) auto
finally show \( \theta \cdot \psi + \theta \cdot x \).

63
The following theorems use summation by parts to relate different prime-counting functions to one another with an integral as a remainder term.

**lemma \( \vartheta \text{-conv-} \pi \text{-integral:} \)**

assumes \( x \geq 2 \)

shows \( \((\lambda t. \pi t / t) \text{ has-integral } (\pi x \cdot \ln x - \vartheta x)\) \{2..x} \)

**proof (cases \( x = 2 \))**

case False

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

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

have \((\lambda t. \text{sum-upto } (\text{ind prime}) t * (1 / t)) \text{ has-integral}

\( \text{sum-upto } (\text{ind prime}) x * \ln x - \text{sum-upto } (\text{ind prime}) 2 * \ln 2 - \)

\( (\sum n \in \text{real} -' \{2<..x\}, \text{ind prime } n * \ln (\text{real } n))\) \{2..x\} using \( x \)

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

(auto simp: continuous-intros derivative-eq-intros

simp flip: has-real-derivative-iff-has-vector-derivative)

hence \((\lambda t. \pi t / t) \text{ has-integral } (\pi x \cdot \ln x -

(\pi 2 * \ln 2 + (\sum n \in \text{real} -' \{2<..x\}, \text{ind prime } n * \ln n)))\) \{2..x\}

by (simp add: \( \vartheta \)-def prime-sum-upto-altdef1 algebra-simps)

also have \( \pi 2 * \ln 2 + (\sum n \in \text{real} -' \{2<..x\}, \text{ind prime } n * \ln n) =

(\sum n \in \text{insert } 2 \text{ (real } -' \{2<..x\}, \text{ind prime } n * \ln n)\)

by (subst sum.insert) (auto simp: eval-\( \vartheta \))

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 \( ?\text{thesis} \).

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

**lemma \( \pi \text{-conv-} \vartheta \text{-integral:} \)**

assumes \( x \geq 2 \)

shows \( \((\lambda t. \vartheta t / (t * \ln t - 2)) \text{ has-integral } (\pi x - \vartheta x / \ln x)\) \{2..x\}

**proof (cases \( x = 2 \))**

case False

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

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

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

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

-(\text{sum-upto } b x * (1 / \ln x) - \text{sum-upto } b 2 * (1 / \ln 2) -

(\sum n \in \text{real} -' \{2<..x\}, b n * (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-real-derivative-iff-has-vector-derivative simp add: power2-eq-square)

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

by (simp add: \( \vartheta \)-def \( \vartheta \)-def prime-sum-upto-altdef1 fun-eq-iff)

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

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

\( \vartheta x * (1 / \ln x) - (\sum n \in \text{insert } 2 \text{ (real } -' \{2<..x\}, b n * (1 / \ln (\text{real } n)))\)

by (subst sum.insert) (auto simp: \( \vartheta \)-def eval-\( \vartheta \))

64
also have $\left( \sum n \in \text{insert } 2 \left( \text{real} - \cdot \{2 < \ldots x\} \right) \cdot b \cdot n \cdot (1 / \ln (\text{real } n)) \right) = \pi \cdot x$ using $x$

unfolding $\pi$-def prime-sum-upto-altdef1 sum-upto-def

proof (intro sum.mono-neutral-cong-left ballI, goal-cases)
  case (3 p)
  hence $p = 1$ by auto
  thus ?case by auto
qed (auto simp: b-def)

finally show ?thesis by simp

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

lemma integrable-weighted-\vartheta:
  assumes $2 \leq a \leq x$
  shows $\left( \lambda t. \vartheta \cdot t / (t \cdot \ln t - 2) \right) \text{integrable-on} \{a..x\}$

proof (cases $a < x$)
  case True
  hence $\left( \lambda t. \vartheta \cdot t / (t \cdot \ln t - 2) \right) \text{integrable-on} \{a..x\}$ using assms

  unfolding \vartheta-def prime-sum-upto-altdef1
  by (intro partial-summation-integrable-strong[where $X = \{\}$ and $f = \lambda x. -1 / \ln x$])

  (auto simp flip: has-real-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-ifff)

lemma \vartheta-conv-M-integral:
  assumes $x \geq 2$
  shows $\left( M \cdot \text{has-integral} \left( M \cdot x \cdot x - \vartheta \cdot x \right) \right) \{2..x\}$

proof (cases $x = 2$)
  case False
  with assms have $x > 2$ by simp

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

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

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

have $\left( \lambda t. \text{sum-upto } b \cdot t \cdot 1 \right) \text{has-integral} \text{sum-upto } b \cdot x \cdot x - \text{sum-upto } b \cdot 2 \cdot 2$

  $\left( \sum n \in \text{real} - \cdot \{2 < \ldots x\} \cdot b \cdot n \cdot \text{real } n \right) \{2..x\}$ using $x$

  by (intro partial-summation-strong[of {}])

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

also have sum-upto $b = M$
  by (simp add: fun-eq_iff primes-M-def b-def prime-sum-upto-altdef1)

also have $M \cdot x \cdot x - M \cdot 2 \cdot 2 - \left( \sum n \in \text{real} - \cdot \{2 < \ldots x\} \cdot b \cdot n \cdot \text{real } n \right) = M \cdot x \cdot x - \left( \sum n \in \text{insert } 2 \left( \text{real} - \cdot \{2 < \ldots x\} \right) \cdot b \cdot n \cdot \text{real } n \right)$

  by (subst sum.insert) (auto simp: eval-M b-def)

also have $\left( \sum n \in \text{insert } 2 \left( \text{real} - \cdot \{2 < \ldots x\} \right) \cdot b \cdot n \cdot \text{real } n \right) = \vartheta \cdot x$
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 \( M \)-conv-\( \vartheta \)-integral:
assumes \( x \ge 2 \)
shows \( (\lambda t. \vartheta t / t^2) \) has-integral \( (M x - \vartheta x / x) \) \{2..\}
proof (cases \( x = 2 \))
case False
with \( \text{assms} \) have \( x > 2 \) by simp
define \( b :: \text{nat} \) \Rightarrow \text{real} \ where \( b = (\lambda p. \text{ind prime } p * \ln p) \)
note [intro] = finite-vimage-real-of-nat-greaterThanAtMost
have prime-le-2: \( p = 2 \) if \( p \le 2 \) prime \( p \) for \( p :: \text{nat} \)
using that by (auto simp: prime-nat-iff)

have \((\lambda t. \text{sum-upto } b \ t * (1 / t^2)) \) has-integral
sum-upto \( b \ \vartheta x * (-1 / x) - \) sum-upto \( b \ 2 * (-1 / 2) - \)
\((\sum n \in \text{real} -' \{2<..\}, \ b \ n * (-1 / \text{real } n)) \) \{2..\} using \( x \)
by (intro partial-summation-strong[of \{\}])
(auto simp flip: has-real-derivative-iff-has-vector-derivative simp: power2-eq-square
intro!: derivative-eq-intros continuous-intros)
also have \( \text{sum-upto } b = \vartheta \)
by (simp add: fun-eq_iff \( \vartheta \)-def b-def prime-sum-upto-altdef1)
also have \( \vartheta x \) \* \((-1 / x) - \vartheta 2 * (-1 / 2) - \)(\sum n \in \text{real} -' \{2<..\}, \ b \ n * (-1 / \text{real } n)) =
\((-\vartheta x / x - (\sum n \ininsert 2 \ (\text{real} -' \{2<..\}), \ b \ n / \text{real } n))\)
by (subst sum.insert) (auto simp: eval-\( \vartheta \) b-def sum-negf)
also have \( (\sum n \ininsert 2 \ (\text{real} -' \{2<..\}), \ b \ n / \text{real } n) = M \ x \)
unfolding primes-M-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 integrable-primes-M: \( M \) integrable-on \{x..y\} if \( 2 \le x \) for \( x \ y :: \text{real} \)
proof —
have \((\lambda x. M \ x * 1) \) integrable-on \{x..y\} if \( 2 \le x < y \) for \( x \ y :: \text{real} \)
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-real-derivative-iff-has-vector-derivative
intro!: derivative-eq-intros continuous-intros)
thus \( \text{thesis} \) using that-has-integral-refl(2)[of \( M \ x \)] by (cases \( x \ y \) rule: linorder-cases)
auto
qed

3.5 Bounds

lemma \( \vartheta \)-upper-bound-coarse:
assumes \( x \ge 1 \)
shows \( \theta \ x \leq x \times \ln x \)

proof –

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

sum-upto-def

by (intro sum-mono) (auto simp: ind-def)
also have \( \ldots \leq \text{real-of-int} \ \lfloor x \rfloor \times \ln x \) using assms

by (simp add: sum-upto-altdef)
also have \( \ldots \leq x \times \ln x \) using assms by (intro mult-right-mono) auto
finally show \( \theta \text{thesis} \).

qed

lemma \( \theta \text{-le-}\psi: \theta \ x \leq \psi \ x \)

proof (cases \( x \geq 2 \))

  case False

  hence \( \lfloor x \rfloor = 0 \lor \lfloor x \rfloor = 1 \) by linarith

  thus \( \theta \text{thesis} \) by (auto simp: eval-\theta)

next

  case True

  hence \( \psi \ x - \theta \ x = (\sum i \ | \ 2 \leq i \land \text{real} \ i \leq \log 2 \ x \ . \ \theta \ (\text{root} \ i \ x)) \)

  by (rule \( \psi \text{-minus-}\theta)\)
also have \( \ldots \geq 0 \) by (intro sum-nonneg) auto
finally show \( \theta \text{thesis} \) by simp

qed

lemma \( \pi \text{-upper-bound-coarse}: \)

  assumes \( x \geq 0 \)

  shows \( \pi \ x \leq x / 3 + 2 \)

proof –

  have \{ \( p \text{. prime} \ p \land p \leq \lfloor x \rfloor \} \subseteq \{2, 3\} \cup \{ \ p \text{. p} \neq 1 \land \text{odd} \ p \land \neg 3 \text{ \text{dvd} } p \land p \leq \lfloor x \rfloor \} \)

also have \( \ldots \subseteq \{2, 3\} \cup \{(\lambda k. 6 \ast k+1) \ \cdot \ \{0 << \lfloor (x+5)/6 \rfloor \} \cup (\lambda k. 6 \ast k+5) \ \cdot \ \{< \lfloor (x+1)/6 \rfloor \})\)

  (is - \cup \ ?lhs \subseteq - \cup ?rhs)

proof (intro Un-mono subsetI)

  fix \( p \) :: nat assume \( p \in \ ?lhs \)

  hence \( p \neq 1 \text{ \ odd} \ p \text{ \ \text{dvd} } p \leq \lfloor x \rfloor \) by auto

  from \( p \ (1-3) \) have \( (3k. k > 0 \land p = 6 \ast k + 1 \lor p = 6 \ast k + 5) \) by presburger

  then obtain \( k \) where \( k > 0 \land p = 6 \ast k + 1 \lor p = 6 \ast k + 5 \) by blast

  hence \( p = 6 \ast k + 1 \land k > 0 \land k \leq \lfloor (x+5)/6 \rfloor \lor p = 6 \ast k + 5 \land k = \lfloor (x+1)/6 \rfloor \)

  unfolding add-divide-distrib using \( p(\acute{\lambda}) \) by linarith

  thus \( p \in \ ?rhs \) by auto

qed

finally have \( \text{subset}: \{ \ p. \ \text{prime} \ p \land p \leq \lfloor x \rfloor \} \subseteq \ldots \ (\text{is - \subseteq \ ?A}) \).

have \( \pi \ x = \text{real} \ (\text{card} \ \{ \ p. \ \text{prime} \ p \land p \leq \lfloor x \rfloor \}) \)

by (simp add: \( \pi \text{-def prime-sum-upto-altdef2} \))
also have \( \text{card} \{ p \text{, prime} \ p \land p \leq \text{nat} \ [x] \} \leq \text{card} \ ?A \)
by (intro card-mono subset) auto
also have \( \ldots \leq 2 + (\text{nat} \ [ (x+5)/6 ] - 1 + \text{nat} \ [ (x+1)/6 ] ) \)
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 assms unfolding add-divide-distrib by (cases x \geq 1, linarith, simp)
finally show \(?thesis by simp\)
qed

lemma le-numeral-iff: \( m \leq \text{numeral} n \rightleftharpoons m = \text{numeral} n \lor m \leq \text{pred-numeral} n \)
using numeral-eq-Suc by presburger

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 \( n > 0 \Longrightarrow F n < 4 \sim n \)
proof (induction \( n \) rule: less-induct)
case (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 \( n = 0 \mid n = 1 \mid n = 2 \mid n = 3 \mid \text{even} \ n \land n \geq 4 \mid \text{odd} \ n \land n \geq 4 \)
by metis
thus \(?case\)
proof cases
assume \[ \text{simp} \]: \( n = 1 \)
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 \[ \text{simp} \]: \( n = 2 \)
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 \[ \text{simp} \]: \( n = 3 \)
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 \( n \): \( \text{even} \ n \land n \geq 4 \)
from \( n \) have \( F \ (n - 1) \ < 4 \sim (n - 1) \) by (intro less.IH) auto
also have \( \text{prime} \ p \land p \leq n \rightleftharpoons \text{prime} \ p \land p \leq n - 1 \) for \( p \)
using \( n \) prime-odd-nat[of \( n \)] by (cases \( p = n \)) auto
hence \( F \ (n - 1) = F \ n \) by (simp add: F-def)
also have \( 4 \sim (n - 1) \leq (4 \sim n :: \text{nat}) \) by (intro power-increasing) auto
finally show \(?case\)
next

68
assume \( n \): odd \( n \geq 4 \)
then obtain \( k \) where \( k \)-eq: \( n = \text{Suc} (2 \ast k) \) by (auto elim: oddE)
from \( n \) have \( k \geq 2 \) unfolding \( k \)-eq by presburger
have prime-dvd: \( p \) dvd \( \{ \text{n choose } k \} \) if \( p: \text{prime} p \) \( p \in \{ \text{k+1<n} \} \) for \( p \)
proof
  from \( p \) \( n \) have \( p \) dvd pochhammer \( (k + 2) \) \( k \)
    unfolding pochhammer-prod
    by (subst prime-dvd-prod-iff)
      (auto intro: bezel[of - \( p - k - 2 \)] simp: k-eq numeral-2-eq-2 Suc-diff-Suc)
also have pochhammer \( (\text{real} (k + 2)) \) \( k = \text{real} ((\text{n choose } k) \ast \text{fact } k) \)
      by (simp add: binomial-gbinomial gbinomial-pochhammer\(^r \) \( k \)-eq field-simps)
  hence pochhammer \( (k + 2) \) \( k = \text{n choose } k \ast \text{fact } k \)
    unfolding pochhammer-of-nat of-nat-eq-iff
finally show \( p \) dvd \( n \) choose \( k \) using \( p \)
  by (auto simp: prime-dvd-fact-iff prime-dvd-mult-nat)
qed

have \( \prod \{ p: \text{prime } p \land p \in \{ k+1..<n \} \} \) dvd \( \{ \text{n choose } k \} \)
proof (rule multiplicity-le-imp-dvd, goal-cases)
  case \( (2 \) \( p \)\)
  thus \( \text{?case} \)
  proof (cases \( p \in \{ k+1..<n \} \))
    case False
    hence multiplicity \( p \) \( \{ \text{\prod \{ p: \text{prime } p \land p \in \{ k+1..<n \} \} \} \} = 0 \) using \( 2 \)
      by (subst prime-multiplicity-prod-distrib) (auto simp: prime-multiplicity-other)
    thus \( \text{?thesis by auto} \)
  next
    case True
    hence multiplicity \( p \) \( \{ \text{\prod \{ p: \text{prime } p \land p \in \{ k+1..<n \} \} \} \} = \sum \text{\{ multiplicity } p \{ p: \text{prime } p \land \text{Suc } k < p \land p \leq n \} \) using \( 2 \)
      by (subst prime-multiplicity-prod-distrib) auto
    also have \( \ldots = \sum \text{\{ multiplicity } p \{ p \} \) using \( \text{True 2} \)
      proof (intro sum.mono-neutral-right ballI)
        fix \( q :: \text{nat} \) assume \( q \in \{ p: \text{prime } p \land \text{Suc } k < p \land p \leq n \} - \{ p \} \)
        hence multiplicity \( p \) \( q = 0 \) using \( 2 \)
          by (cases \( p = q \)) (auto simp: prime-multiplicity-other)
    qed auto
    also have \( \ldots = 1 \) using \( 2 \) by simp
    also have \( 1 \leq \text{multiplicity } p \) \( \{ \text{n choose } k \} \)
      using prime-dvd[of \( p \)] \( 2 \) \( \text{True by (intro multiplicity-geI)} \) auto
    finally show \( \text{?thesis} \).
  qed auto
qed

hence \( \prod \{ p: \text{prime } p \land p \in \{ k+1..<n \} \} \leq \{ n \text{ choose } k \} \)
  by (intro dvd-imp-le) (auto simp: k-eq)
also have \( \ldots = 1 / 2 \ast (\sum i \in \{ k, \text{Suc } k \}. \text{n choose } i) \)
  using central-binomial-odd[of \( n \)] by (simp add: k-eq)
also have \( (\sum i \in \{ k, \text{Suc } k \}. \text{n choose } i) < (\sum i \in \{ 0, k, \text{Suc } k \}. \text{n choose } i) \)
  using \( k \) by simp

69
also have ... ≤ (∑ i≤n. n choose i)
  by (intro sum-mono2) (auto simp: k-eq)
also have ... = (1 + 1) ^ n
  using binomial[of 1 1 n] by simp
also have 1 / 2 * ... = real (4 ^ k)
  by (simp add: k-eq power-mult)
finally have less: (∏{p. prime p ∧ p ∈ {k + 1..n}}) < 4 ^ k
  unfolding of-nat-less-iff by simp
have F n = F (Suc k) * (∏{p. prime p ∧ p ∈ {k+1..n}}) unfolding F-def
  by (subst prod.union-disjoint [symmetric]) (auto intro!: prod.cong simp: k-eq)
also have ... < 4 ^ Suc k * 4 ^ k using n
  by (intro mult-strict-mono less less.IH) (auto simp: k-eq)
also have ... = 4 ^ (Suc k + k)
  by (simp add: power-add)
also have Suc k + k = n by (simp add: k-eq)
finally show ?case.
qed (insert less.prems, auto)
qed

lemma ϑ-upper-bound:
  assumes x: x ≥ 1
  shows ϑ x < ln 4 * x
proof -
  have 4 powr (ϑ x / ln 4) = (∏ p | prime p ∧ p ≤ nat [x]. 4 powr (log 4 (real p)))
    by (simp add: ϑ-def powr-sum prime-sum-upto-altdef2 sum-divide-distrib log-def)
  also have ... < 4 ^ nat [x]
    using x
  finally have 4 powr (ϑ x / ln 4) < 4 ^ nat [x]
    thus ϑ x < ln 4 * x
    by (subst (asm) powr-less-cancel-iff) (auto simp: field-simps)
qed

lemma ϑ-bigo: ϑ ∈ O(λx. x)
  by (intro le-imp-bigo-real[of ln 4] eventually-mono[of eventually-ge-at-top[of 1]]
    less-imp-le[of ϑ-upper-bound]) auto

lemma ψ-minus-ϑ-bound:
  assumes x: x ≥ 2
shows \( \psi x - \vartheta x \leq 2 \ln x \sqrt{x} \)

proof –

have \( \psi x - \vartheta x = \left( \sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 \ x. \vartheta (\text{root } i \ x) \right) \) using \( x \)

by (rule \( \psi \)-\( \vartheta \)-bigo)

also have \( \ldots \leq \left( \sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 \ \ln 4 \wedge \text{root } i \ x \right) \) using \( x \) by (intro sum-mono less-\( \vartheta \)-impl-\( \vartheta \)-upper-bound) auto

also have \( \ldots \leq \left( \sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 \ \ln 4 \wedge \text{root } 2 \ x \right) \) using \( x \)

by (intro sum-mono mult-mono) (auto simp: \( \vartheta \)-log-iff power-realpow intro!:
real-root-decreasing)

also have \( \ldots = \text{card } \{ i. \ 2 \leq i \wedge \text{real } i \leq \log 2 \ x \} \ast \ln 4 \ast \sqrt{x} \)

by (simp add: sqrt-def)

also have \( \{ i. \ 2 \leq i \wedge \text{real } i \leq \log 2 \ x \} \ast \text{card } \{ \text{2..nat } | \log 2 \ x \} \)

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 \( (\text{nat } | \log 2 \ x - 1) \leq \log 2 \ x \) using \( x \) by linarith

hence \( \text{card } \{ \text{2..nat } | \log 2 \ x \} \leq \log 2 \ x \) by simp

also have \( \ln (2 \ast 2 :: \text{real}) = 2 \ast \ln 2 \) by (subst ln-mult) auto

hence \( \log 2 \ x \ast \ln 4 \ast \sqrt{x} = 2 \ast \ln x \ast \sqrt{x} \) using \( x \)

by (simp add: ln-sqrt log-def power2-eq-square field-simps)

finally show \( ?\text{thesis} \) using \( x \) by (simp add: mult-right mono)

qed

lemma \( \psi \)-\( \vartheta \)-bigo: \( (\lambda x. \psi x - \vartheta x) \in O(\lambda x. \ln x \ast \sqrt{x}) \)

proof (intro bigoI [of 2] eventually mono [OF eventually ge at top [of 2]])

fix \( x :: \text{real} \) assume \( x \geq 2 \)

thus \( \text{norm } (\psi x - \vartheta x) \leq 2 \ast \text{norm } (\ln x \ast \sqrt{x}) \)

using \( \psi \)-\( \vartheta \)-bound [of \( x \) \( \vartheta \)-le \( \vartheta \)-bound [of \( x \) \( \vartheta \)-le \psi [of \( x \)]] by simp

qed

lemma \( \psi \)-bigo: \( \psi \in O(\lambda x. x) \)

proof –

have \( (\lambda x. \psi x - \vartheta x) \in O(\lambda x. \ln x \ast \sqrt{x}) \)

by (rule \( \psi \)-\( \vartheta \)-bigo)

also have \( (\lambda x. \ln x \ast \sqrt{x}) \in O(\lambda x. x) \)

by real-asym

finally have \( (\lambda x. \psi x - \vartheta x + \vartheta x) \in O(\lambda x. x) \)

by (rule sum-in bigo) (fact \( \vartheta \)-bigo)

thus \( ?\text{thesis} \) by simp

qed

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 $\int \{2..x\} (\lambda t. 1 / \ln t ^ 2) \leq \sqrt{x} / \ln 2 ^ 2 + 4 \times x / \ln x ^ 2$
proof –
from $x$ have $x \times 1 \leq x ^ 2$ unfolding power2-eq-square by (intro mult-left-mono)
with $x$ have $x': 2 \leq \sqrt{x} \times \sqrt{x} \leq x$
by (auto simp: real-sqrt-le-iff intro!: real-le-rsqrt)
also have $\exists x': \leq \sqrt{x} / \ln 2 ^ 2$ using $x'$ by (simp add: field-simps)
also have $\exists x': \leq \sqrt{x} / \ln 2 ^ 2$ using $x'$ by (intro Henstock-Kurzweil-Integration.integrate-continuous-real)
also have $\exists x': \leq \sqrt{x} / \ln 2 ^ 2$ using $x'$
by (intro Henstock-Kurzweil-Integration.integrate-continuous-real)
finally show $?thesis$ by simp
qed

**lemma** integral-one-over-log-squared-bigo:
$(\lambda x::real. \int \{2..x\} (\lambda t. 1 / \ln t ^ 2)) \in O(\lambda x. x / \ln x ^ 2)$
proof –
define $ab$ where $ab = (\lambda x::real. \sqrt{x} / \ln 2 ^ 2 + 4 \times x / \ln x ^ 2)$
have eventually $(\lambda x. \int \{2..x\} (\lambda t. 1 / (\ln t ^ 2)) \leq |ub x|)$ at-top
using eventually-ge-at-top[of $4$]
proof eventually-elim
case $(elim x)$
hence $\int \{2..x\} (\lambda t. 1 / \ln t ^ 2) = \int \{2..x\} (\lambda t. 1 / \ln t ^ 2)$
by (intro abs-of-nonneg integrate-continuous-real)
also have $(\lambda x. \int \{2..x\} (\lambda t. 1 / (\ln t ^ 2)) \in O(\lambda x. x / \ln x ^ 2)$ unfolding $ub$ by real-asym
finally show $?thesis$ .
qed

**lemma** $\pi$-bound:
assumes $x \geq (\frac{4}{3} :: \text{real})$
defines $\text{ub} \equiv 2 / \ln 2 * \sqrt{x} + 8 * \ln 2 * x / \ln x ^ {2}$
shows $\pi x - \vartheta x / \ln x \in \{0..\text{ub}\}$

proof
- define $r$ where $r = (\lambda x. \text{integral} \{2..x\} (\lambda t. \vartheta t / (t * \ln t ^ {2})))$
have integrable: $(\lambda c / \ln t ^ {2}) \text{integrable-on} \{2..x\} for c$
  by (intro integrable-continuous-real continuous-intros) auto

have $r x \leq \text{integral} \{2..x\} (\lambda t. \ln 4 / \ln t ^ {2})$ unfolding r-def
  using integrable-weighted-\vartheta[of 2 x] integrable[of ln 4] assms less-imp-le[OF \vartheta-upper-bound]
    by (intro integral-le divide-right-mono) (auto simp: field-simps)
also have $\ldots = \ln 4 * \text{integral} \{2..x\} (\lambda t. 1 / \ln t ^ {2})$
  using integrable[of 1] by (subst integral-mult) auto
also have $\ldots \leq \ln 4 * (\sqrt{x} / \ln 2 ^ {2} + 4 * x / \ln x ^ {2})$
  using assms by (intro mult-left-mono integral-one-over-log-squared-bound) auto
also have $\ln (\frac{4}{3} :: \text{real}) = 2 * \ln 2$
  using ln-realpow[of 2 2] by simp
also have $\ldots * (\sqrt{x} / \ln 2 ^ {2} + 4 * x / \ln x ^ {2}) = \text{ub}$
  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 $r x \in \{0..\text{ub}\}$ by auto
with $\pi \text{-conv-\vartheta-integral[of} x\text{]} \text{assms(1)}$ 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.

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

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

lemma $\pi$-$\text{upper-bound}$:
assumes \( x \geq \left( \frac{4}{:	ext{real}} \right) \)
shows \( \pi \ x < \ln 4 * \ x / \ln x + 8 * \ln 2 * \ x / \ln x \sim 2 + 2 / \ln 2 * \sqrt{x} \)
proof –
define \( ub \) where \( ub = 2 / \ln 2 * \sqrt{x} + 8 * \ln 2 * \ x / \ln x \sim 2 \)
have \( \pi \ x \leq \vartheta \ x / \ln x + ub \)
also from \( \text{assms} \) have \( \vartheta \ x / \ln x < \ln 4 * \ x / \ln x \)
by \( \text{(intro } \vartheta-\text{-upper-bound divide-strict-right-mono) auto} \)
finally show \( \text{thesis} \)
using \( \text{assms by (simp add: algebra-simps ub-def)} \)
qed

lemma \( \pi\text{-bigo}: \pi \in O(\lambda x. x / \ln x) \)
proof –
have \( (\lambda x. \pi \ x - \vartheta \ x / \ln x) \in O(\lambda x. x / \ln x \sim 2) \)
by \( \text{(fact } \pi\text{-}\vartheta\text{-bigo)} \)
also have \( (\lambda x:\text{real. } x / \ln x \sim 2) \in O(\lambda x. x / \ln x) \)
by \( \text{real-asymp} \)
finally have \( (\lambda x. \pi \ x - \vartheta \ x / \ln x) \in O(\lambda x. x / \ln x) \)
moreover have eventually \( (\lambda x:\text{real. } x \neq 0) \text{-at-top by real-asymp} \)
hence eventually \( (\lambda x::\text{real. } ln x \neq 0) \text{-at-top by eventually-elim auto} \)
hence \( (\lambda x. \vartheta \ x / \ln x) \in O(\lambda x. x / \ln x) \)
using \( \vartheta\text{-bigo by (intro landau-o.big.divide-right)} \)
ultimately have \( (\lambda x. \pi \ x - \vartheta \ x / \ln x + \vartheta \ x / \ln x) \in O(\lambda x. x / \ln x) \)
by \( \text{(rule sum-in-bigo)} \)
thus \( \text{thesis by simp} \)
qed

3.6 Equivalence of various forms of the Prime Number Theorem

In this section, we show that the following forms of the Prime Number Theorem are all equivalent:

1. \( \pi(x) \sim x / \ln x \)
2. \( \pi(x) \ln \pi(x) \sim x \)
3. \( p_n \sim n \ln n \)
4. \( \vartheta(x) \sim x \)
5. \( \psi(x) \sim x \)

We show the following implication chains:

- \( (1) \rightarrow (2) \rightarrow (3) \rightarrow (2) \rightarrow (1) \)
- \( (1) \rightarrow (4) \rightarrow (1) \)
• (4) → (5) → (4)

All of these proofs are taken from Apostol’s book.

**lemma PNT1-imp-PNT1':**

*assumes* $\pi \sim_{[at-top]} (\lambda x. x / \ln x)$

*shows* $(\lambda x. \ln (\pi x)) \sim_{[at-top]} \ln$

*proof –*

*from asms* **have** $(\lambda x. \pi x / (x / \ln x)) \longrightarrow 1$ *at-top*

*by* (rule **asymp-equiv3-strong** [OF - eventually-mono] [OF eventually-gt-at-top [of 1]])

**auto**

*also have* $(\lambda x. \ln (\pi x) - \ln x + \ln (\ln x)) \longrightarrow 0$ *at-top*

*by* (intro filterlim-cong eventually-mono [OF eventually-gt-at-top [of 2]])

(auto simp: ln-field-simps ln-mult pi-pos)

**finally show** $(\lambda x. \ln (\pi x) - \ln x + \ln (\ln x)) \in o(\lambda x. 1)$

*by* (intro smalloI-tendsto)

**also have** $(\lambda : real. 1 :: real) \in o(\lambda x. \ln x)$

*by* real-asymp

**finally have** $(\lambda x. \ln (\pi x) - \ln x + \ln (\ln x) - \ln (\ln x)) \in o(\lambda x. \ln x)$

*by* (rule sum-in-smallo) real-asymp+

**thus**: $(\lambda x. \ln (\pi x)) \sim_{[at-top]} \ln$

*by* (simp add: **asymp-equiv-altdef**)

**qed**

**lemma PNT1-imp-PNT2:**

*assumes* $\pi \sim_{[at-top]} (\lambda x. x / \ln x)$

*shows* $(\lambda x. \pi x * \ln (\pi x)) \sim_{[at-top]} (\lambda x. x)$

*proof –*

**have** $(\lambda x. \pi x * \ln (\pi x)) \sim_{[at-top]} (\lambda x. x / \ln x * \ln x)$

*by* (intro **asymp-equiv-intros** asms PNT1-imp-PNT1')

**also have** $\ldots \sim_{[at-top]} (\lambda x. x)$

*by* (intro **asymp-equiv-refl-ev** eventually-mono [OF eventually-gt-at-top [of 1]])

(auto simp: **field-simps**)

**finally show** $(\lambda x. \pi x * \ln (\pi x)) \sim_{[at-top]} (\lambda x. x)$

*by* simp

**qed**

**lemma PNT2-imp-PNT3:**

*assumes* $(\lambda x. \pi x * \ln (\pi x)) \sim_{[at-top]} (\lambda x. x)$

*show* $(\text{nth-prime} ~_{[at-top]} (\lambda n. n * \ln n))$

*proof –*

**have** $(\lambda n. \text{nth-prime} n) \sim_{[at-top]} (\lambda n. \pi (\text{nth-prime} n) * \ln (\pi (\text{nth-prime} n)))$

*using asms*

*by* (rule **asymp-equiv-symI** [OF **asymp-equiv-compose’**])

(auto intro: filterlim-compose [OF filterlim-real-sequentially nth-prime-at-top])

**also have** $\ldots = (\lambda n. \real (\text{Suc} n) * \ln (\real (\text{Suc} n)))$

*by* (simp add: add-ac)

75
also have \ldots \sim_{\text{at-top}} (\lambda n. \text{real } n \ast \ln (\text{real } n))

by \text{real-asymp}

finally show nth-prime \sim_{\text{at-top}} (\lambda n. \text{real } n) .

definitions

\text{lemma PNT3-imp-PNT2:}

\text{assumes nth-prime \sim_{\text{at-top}} (\lambda n. \text{real } n)}

\text{shows } (\lambda x. \pi x \ast \ln (\pi x)) \sim_{\text{at-top}} (\lambda x. x)

\text{proof (rule asymp-equiv-sym, rule asymp-equiv-sandwich-real)}

\text{show eventually } (\lambda x. x \in \{ \text{real } (\text{nth-prime } (\text{nat } \lfloor \pi x \rfloor - 1))..\text{real } (\text{nth-prime } (\text{nat } \lfloor \pi x \rfloor))\})

\text{at-top}

\text{using eventually-ge-at-top[of 2]}

\text{proof eventually-elim}

\text{case (elim x)}

\text{with nth-prime-partition"[of x] show } \text{?case by auto}

\text{qed}

\text{next}

\text{have } (\lambda x. \text{real } (\text{nth-prime } (\text{nat } \lfloor \pi x \rfloor - 1)) ) \sim_{\text{at-top}}

(\lambda x. \text{real } (\text{nat } \lfloor \pi x \rfloor - 1) \ast \ln (\text{real } (\text{nat } \lfloor \pi x \rfloor - 1)))

\text{by (rule asymp-equiv-compose"[OF - \pi-at-top], rule asymp-equiv-compose"[OF assms]) real-asymp}

\text{also have \ldots \sim_{\text{at-top}} (\lambda x. \pi x \ast \ln (\pi x))}

\text{by (rule asymp-equiv-compose"[OF - \pi-at-top]) real-asymp}

\text{finally show } (\lambda x. \text{real } (\text{nth-prime } (\text{nat } \lfloor \pi x \rfloor - 1))) \sim_{\text{at-top}} (\lambda x. \pi x \ast \ln (\pi x)) .

\text{next}

\text{have } (\lambda x. \text{real } (\text{nth-prime } (\text{nat } \lfloor \pi x \rfloor))) \sim_{\text{at-top}}

(\lambda x. \text{real } (\text{nat } \lfloor \pi x \rfloor) \ast \ln (\text{real } (\text{nat } \lfloor \pi x \rfloor)))

\text{by (rule asymp-equiv-compose"[OF - \pi-at-top], rule asymp-equiv-compose"[OF assms]) real-asymp}

\text{also have \ldots \sim_{\text{at-top}} (\lambda x. \pi x \ast \ln (\pi x))}

\text{by (rule asymp-equiv-compose"[OF - \pi-at-top]) real-asymp}

\text{finally show } (\lambda x. \text{real } (\text{nth-prime } (\text{nat } \lfloor \pi x \rfloor))) \sim_{\text{at-top}} (\lambda x. \pi x \ast \ln (\pi x)) .

\text{qed}

\text{lemma PNT2-imp-PNT1:}

\text{assumes } (\lambda x. \pi x \ast \ln (\pi x)) \sim_{\text{at-top}} (\lambda x. x)

\text{shows } (\lambda x. \ln (\pi x)) \sim_{\text{at-top}} (\lambda x. \ln x)

\text{and } \pi \sim_{\text{at-top}} (\lambda x. x / \ln x)

\text{proof --}

\text{have ev: eventually } (\lambda x. \pi x > 0) \text{-at-top}

\text{eventually } (\lambda x. \ln (\pi x) > 0) \text{-at-top}

\text{eventually } (\lambda x. \ln (\ln (\pi x)) > 0) \text{-at-top}

\text{by (rule eventually-compose-filterlim[OF - \pi-at-top], real-asymp)+}

\text{let } \lambda f = \lambda x. 1 + \ln (\ln (\pi x)) / \ln (\pi x) - \ln x / \ln (\pi x)

\text{have } ((\lambda x. \ln (\pi x) \ast \lambda f x) \longrightarrow \ln 1) \text{-at-top}

\text{proof (rule Lim-transform-eventually)}
from assms have \((\lambda x. \pi x * \ln (\pi x) / x) \longrightarrow 1\) at-top
by (rule asymp-equivD-strong[OF eventually-mono[OF eventually-gt-at-top[of 1]]) auto

then show \((\lambda x. \ln (\pi x * \ln (\pi x) / x)) \longrightarrow \ln 1\) at-top
by (rule tendsto-ln auto)
show \(\forall x \in\) at-top. \(\ln (\pi x * \ln (\pi x) / x) = \ln (\pi x) * ?f x\)
using eventually-gt-at-top[of 0] ev
by eventually-elim (simp add: field-simps ln-mul ln-div)

qed

moreover have \((\lambda x. 1 / \ln (\pi x)) \longrightarrow 0\) at-top
by (rule filterlim-compose[OF OF - \pi-at-top]) real-asymp

ultimately have \((\lambda x. \ln (\pi x) * ?f x * (1 / \ln (\pi x))) \longrightarrow \ln 1 * 0\) at-top
by (rule tendsto-mul)

moreover have eventually (\(\lambda x. \ln (\pi x) * ?f x * (1 / \ln (\pi x)) = ?f x\)) at-top
using ev by eventually-elim auto

ultimately have \(?f \longrightarrow \ln 1 * 0\) at-top
by (rule Lim-transform-eventually)

hence \((\lambda x. 1 + \ln (\ln (\pi x)) / \ln (\pi x) - ?f x) \longrightarrow 1 + 0 - \ln 1 * 0\) at-top
by (intro tendsto-intros filterlim-compose[OF OF - \pi-at-top]) (real-asymp | simp)+

hence \((\lambda x. \ln x / \ln (\pi x)) \longrightarrow 1\) at-top
by simp

thus \(\ast: (\lambda x. \ln (\pi x) \sim [\text{at-top}] (\lambda x. \ln x)\)
by (rule asymp-equiv-sym[OF OF asymp-equivI])

have eventually \((\lambda x. \pi x = \pi x * \ln (\pi x) / \ln (\pi x))\) at-top
using ev by eventually-elim auto

hence \(\pi \sim [\text{at-top}] (\lambda x. \pi x * \ln (\pi x) / \ln (\pi x))\)
by (rule asymp-equiv-refl-ev)
also from assms and \(\ast\) have \((\lambda x. \pi x * \ln (\pi x) / \ln (\pi x)) \sim [\text{at-top}] (\lambda x. x / \ln x)\)
by (rule asymp-equiv-intros)
finally show \(\pi \sim [\text{at-top}] (\lambda x. x / \ln x)\).

qed

lemma \(\text{PNT}^\sharp\text{-imp-PNT}^5\):
assumes \(\varnothing \sim [\text{at-top}] (\lambda x. x)\)
shows \(\psi \sim [\text{at-top}] (\lambda x. x)\)
proof –
define \(r\) where \(r = (\lambda x. \psi x - \varnothing x)\)
have \(r \in O(\lambda x. \ln x * \sqrt{x})\)
unfolding r-def by (fact \(\psi\)-minus-\(\varnothing\)-bigo)
also have \((\lambda x::\text{real} \cdot \ln x * \sqrt{x}) \in o(\lambda x. x)\)
by real-asymp
finally have \(r: r \in o(\lambda x. x)\).

have \((\lambda x. \varnothing x + r x) \sim [\text{at-top}] (\lambda x. x)\)
using assms \(r\) by (subst asymp-equiv-add-right) auto
thus \(?\text{thesis}\) by (simp add: r-def)

qed
lemma PNT4-imp-PNT1:
assumes \( \vartheta \sim_{[at-top]} (\lambda x. x) \)
shows \( \pi \sim_{[at-top]} (\lambda x. x / \ln x) \)
proof
  have \( (\lambda x. (\pi x - \vartheta x / \ln x)) \in o(\lambda x. x / \ln x) \)
  proof (rule sum-in-smallo)
    have \( (\lambda x. \pi x - \vartheta x / \ln x) \in O(\lambda x. x / \ln x) \)
      by (rule \( \pi \)-\( \vartheta \)-bigo)
    also have \( (\lambda x. x / \ln x) \in o(\lambda x. x / \ln x) \)
      by \( \text{real-asymp} \)
  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 \( \text{real-asymp} \)
  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 \( \text{asymp-equiv-imp} \)-\( \text{bigtheta} \)-\( \text{bigo} \)-\( \text{smallo} \) \( \text{assms} \))
  qed
  thus \( \text{thesis} \) by (simp add: \( \text{diff} \)-\( \text{divide} \)-\( \text{distrib} \) \( \text{asymp-equiv} \)-\( \text{altdef} \))
  qed

lemma PNT1-imp-PNT4:
assumes \( \pi \sim_{[at-top]} (\lambda x. x / \ln x) \)
shows \( \vartheta \sim_{[at-top]} (\lambda x. x) \)
proof
  have \( \vartheta \sim_{[at-top]} (\lambda x. \pi x * \ln x) \)
  proof (rule \( \text{smallo-imp} \)-\( \text{asymp-equiv} \))
    have \( (\lambda x. \vartheta x - \pi x * \ln x) \in \Theta(\lambda x. - ((\pi x - \vartheta x / \ln x) * \ln x)) \)
      by (intro \( \text{bithetaI} \)-\( \text{cong} \) \( \text{eventually-mono} \)[OF \( \text{eventually-gt-at-top} \)[of \( \text{1} \)]])
      (auto simp: \( \text{field-simps} \))
    also have \( (\lambda x. - ((\pi x - \vartheta x / \ln x) * \ln x)) \in O(\lambda x. x / (\ln x)^2 * \ln x) \)
      unfolding \( \text{landau-o.big} \)-\( \text{uminus-in-iff} \) by (intro \( \text{landau-o.big} \)-\( \text{mult-right} \) \( \pi \)-\( \vartheta \)-\( \text{bigo} \))
    also have \( (\lambda x::\text{real}. x / (\ln x)^2 * \ln x) \in o(\lambda x. x / \ln x * \ln x) \)
      by \( \text{real-asymp} \)
    also have \( (\lambda x. x / \ln x * \ln x) \in \Theta(\lambda x. \pi x * \ln x) \)
      by (intro \( \text{asymp-equiv-imp} \)-\( \text{bitheta} \) \( \text{asymp-equiv-intros} \) \( \text{asymp-equiv-symI} \)[OF \( \text{assms} \)])
    finally show \( (\lambda x. \vartheta x - \pi x * \ln x) \in o(\lambda x. \pi x * \ln x) \).
  qed
  moreover have \( \ldots \sim_{[at-top]} (\lambda x. x / \ln x * \ln x) \)
    by (intro \( \text{asymp-equiv-intros} \) \( \text{assms} \))
  moreover have \( \ldots \sim_{[at-top]} (\lambda x. x) \)
    by \( \text{real-asymp} \)
  finally show \( \text{thesis} \).
  qed

lemma PNT5-imp-PNT4:
assumes \( \varphi \sim_{[at-top]} (\lambda x. x) \)
shows \( \theta \sim_{[at-top]} (\lambda x. x) \)
proof

define \( r \) where \( r = (\lambda x. \vartheta x - \psi x) \)

have \((\lambda x. \psi x - \vartheta x) \in O(\lambda x. \ln x \ast \sqrt x)\)
  by \( (\text{fact \( \psi - \vartheta \ast \text{bigo} \))} \)

also have \((\lambda x. \psi x - \vartheta x) = (\lambda x. -r x)\)
  by \( (\text{simp add: \( r \)-def}) \)

finally have \( r \in O(\lambda x. \ln x \ast \sqrt x)\)
  by \( \text{simp} \)

also have \((\lambda x :: \text{real. } \ln x \ast \sqrt x) \in o(\lambda x. x)\)
  by \( \text{real-asym} \)

finally have \( r: r \in o(\lambda x. x)\).

have \((\lambda x. \psi x + r x) \sim [\text{at-top}] (\lambda x. x)\)
  using \( \text{assms \( r \)} \) by \( (\text{subst \( \text{asymp-equiv-add-right} \ast \text{auto}} \)

thus \( \text{thesis by \( (\text{simp add: \( r \)-def}) \)} \)

qed

3.7 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 loga-
rithm of the factorial, namely:

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

We show this using summation by parts.

lemma \( \text{stirling-weak:} \)

assumes \( x: x \geq 1 \)

shows \( \text{sum-upto } \ln x \in \{x \ast \ln x - x - \ln x + 1..x \ast \ln x\} \)

proof (cases \( x = 1 \))

case True

have \( \{0..<\text{Suc } 0\} = \{1\} \) by auto

with True show \( \text{thesis by \( (\text{simp add: sum-upto-altdef}) \)} \)

next

case False

with \( \text{assms have } x: x > 1 \) by simp

have \(((\lambda t. \text{sum-upto } (\lambda x. 1) t \ast (1 / t)) \text{ has-integral}
    \text{sum-upto } (\lambda x. 1) x \ast \ln x - \text{sum-upto } (\lambda x. 1) 1 \ast \ln 1 -
    (\sum \{n \in \text{real -'} \{1..<x\}. 1 \ast \ln (\text{real } n)\}) \{1..x\}) \) \( \text{using} \)
by \( (\text{intro partial-summation-strong[\{\}]}) \)
(\text{auto \( \text{simp flip: has-real-derivative-iff-has-vector-derivative} \)}

79
\begin{verbatim}

lemmas [derivative eq intros continuous intros]

hence \((\lambda t. \text{real} \ \text{nat} \ \{t\} / t) \ \text{has-integral}
\begin{align*}
\text{real} \ \text{nat} \ \{x\} \ast \ln \ x \ast (\sum_{n \in \text{real} \ |- \ \{1..x\}} \ln (\text{real} \ n)) \ \{1..x\}
\end{align*}
\text{by} \ (\text{simp add: sum upto aldef})
\text{also have} \ (\sum_{n \in \text{real} \ |- \ \{1..x\}} \ln (\text{real} \ n)) = \text{sum upto} \ \ln \ x \ \text{unfolding sum upto def}
\text{by} \ (\text{intro sum mono neutral left})
(\text{auto intro: finite subset \{OF - finite vimage real of nat greater than at most\[of 0 \ x]\}})
\text{finally have} \ (*: (\lambda t. \text{real} \ \text{nat} \ \{t\} / t) \ \text{has integral} \ \{x\} \ast \ln \ x - \text{sum upto} \ \ln \ x \ \{1..x\}
\text{using} \ x \ \text{by} \ \text{simp}
\text{have} \ 0 \ \leq \ \text{real of int} \ \{x\} \ast \ln \ x - \text{sum upto} \ (\lambda n. \ln \ \text{real} \ n) \ \{x\}
\text{using} \ * \ \text{by} \ (\text{rule has integral nonneg}) \ \text{auto}
\text{also have} \ \ldots \ \text{\leq} \ x \ast \ln \ x - \text{sum upto} \ \ln \ x
\text{using} \ x \ \text{by} \ (\text{intro diff mono mult mono}) \ \text{auto}
\text{finally have} \ \text{upper: sum upto} \ \ln \ x \ \leq \ x \ast \ln \ x \ \text{by} \ \text{simp}
\text{have} \ (x - 1) \ast \ln \ x - x + 1 \ \leq \ \{x\} \ast \ln \ x - x + 1
\text{using} \ x \ \text{by} \ (\text{intro diff mono mult mono add mono}) \ \text{auto}
\text{also have} \ ((\lambda t. 1) \ \text{has integral} \ \{x - 1\} \ \{1..x\}
\text{using} \ \text{has integral const real \{of 1:: real \ 1 \ x\}} \ \text{by} \ \text{simp}
\text{from} \ * \ \text{and this have} \ \{x\} \ast \ln \ x - \text{sum upto} \ \ln \ x \ \leq \ x - 1
\text{by} \ (\text{rule has integral le}) \ \text{auto}
\text{hence} \ \{x\} \ast \ln \ x - x + 1 \ \leq \ \text{sum upto} \ \ln \ x
\text{by} \ \text{simp}
\text{finally have} \ \text{sum upto} \ \ln \ x \ \geq \ x \ast \ln \ x - x - \ln \ x + 1
\text{by} \ (\text{simp add: algebra simps})
\text{with} \ \text{upper show} \ ?\text{thesis} \ \text{by} \ \text{simp}
\text{qed}
\end{verbatim}

\text{lemma} \ \text{stirling weak bigo:} \ (\lambda x::\text{real} \ \text{sum upto} \ \ln \ x - x \ast \ln \ x) \ \in \ O(\lambda x. x)
\text{proof}
\text{have} \ (\lambda x. \text{sum upto} \ \ln \ x - x \ast \ln \ x) \ \in \ O(\lambda x. -(\text{sum upto} \ \ln \ x - x \ast \ln \ x))
\text{by} \ (\text{subst landau o biguminus}) \ \text{auto}
\text{also have} \ (\lambda x. -(\text{sum upto} \ \ln \ x - x \ast \ln \ x)) \ \in \ O(\lambda x. x + \ln \ x - 1)
\text{proof} \ (\text{intro le imp bigo real of 2\ eventually mono \ OF eventually ge at top of 1}], goal cases)
\text{case} \ (2 x)
\text{thus} \ ?\text{case using} \ \text{stirling weak of x} \ \text{by} \ (\text{auto simp: algebra simps})
\text{next}
\text{case} \ (3 x)
\text{thus} \ ?\text{case using} \ \text{stirling weak of x} \ \text{by} \ (\text{auto simp: algebra simps})
\text{qed auto}
\text{also have} \ (\lambda x. x + \ln x - 1) \ \in \ O(\lambda x::\text{real} x) \ \text{by} \ \text{real asymp}
\text{finally show} \ ?\text{thesis} .
\text{qed}
}\end{verbatim}
The key to showing Mertens’ first theorem is the function

\[ h(x) := \sum_{n \leq x} \frac{\Lambda(n)}{n} \]

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 \( \Re(x) \), as the difference between them is bounded by a constant.

**Lemma** sum-upto-mangoldt-over-id-minus-phi-bounded:

\((\lambda x. \text{sum-upto} \ (\lambda d. \text{mangoldt} \ d \ / \ \text{real} \ d) \ x - \Re x) \in O(\lambda x. 1)\)

**Proof**

1. **Define** \( f \) where \( f = (\lambda d. \text{mangoldt} \ d \ / \ \text{real} \ d) \)
2. **Define** \( C \) where \( C = (\sum p. \ ln \ (\text{real} (p + 1)) \ast (1 / \ \text{real} \ (p \ast (p - 1)))) \)
3. **Prove** \( \text{summable} (\lambda p: \text{nat}. \ ln \ (p + 1) \ast (1 / (p \ast (p - 1)))) \)
4. **Using** \( \text{summable}-\text{comparison-test-big} \)
5. **Define** \( S \) where \( S = \{ (p, i). \ \text{prime} \ p \land 0 < i \land \text{real} \ (p \ ^\ ^\ast i) \leq x \} \)
6. **Prove** \( \text{finite} \ S \) by \( \text{rule finite-subset} \)
7. **Note** \( \text{fin} = \text{finite-subset} \{ \text{OF} \ - \ this, \ unfolded \ S\text{-def} \} \)
8. **Also have** \( S = \{ p. \ \text{prime} \ p \land p \leq x \} \times \{ 1 \} \cup \{ (p, i). \ \text{prime} \ p \land 1 < i \land \text{real} \ (p \ ^\ ^\ast i) \leq x \} \)
9. **By** \( \text{auto simp: Suc-less-eq} \)
10. **Also have** \( (\sum (p, i) | \text{ln} \ (\text{real} \ p) / \ \text{real} \ (p \ ^\ ^\ast i)) = \)

\[ (\sum (p, i) | \text{prime} \ p \land \text{of-nat} \ p \leq x) \times \{ 1 \} \]. \( \text{ln} \ (\text{real} \ p) / \ \text{real} \ (p \ ^\ ^\ast i) \)

\[ (\text{is - = ?S1 + ?S2}) \]
Next, we show that our \( h(x) \) itself is close to \( \ln x \), i.e.:

\[
\sum_{n \leq x} \frac{A(d)}{d} = \ln x + O(1)
\]

**lemma** sum-upto-mangoldt-over-id-asymptotics:
\((\lambda x. \text{sum-upto } (\lambda d. \text{mangoldt } d / \text{real } d) x - \text{ln } x) \in O(\lambda x. 1)\)

\[\text{proof }\]
- \text{define } r \text{ where } r = (\lambda n::\text{real}. \text{sum-upto } (\lambda d. \text{mangoldt } d * (n / d - \text{real-of-int } [n / d])) n)
- \text{have } r : r \in O(\psi)
- \text{proof (intro landau-o.bigl[of 1] eventually-mono[OF eventually-ge-at-top[of 0]])}
  - \text{fix } x :: \text{real assume } x : x \geq 0
  - \text{have eq: } \{1..nat \ [x]\} = \{0<..nat \ [x]\} \text{ by auto}
  - \text{hence } r x \geq 0 \text{ unfolding } r-def \text{ sum-upto-def}
    - \text{by (intro sum-nonneg mult-nonneg-nonneg mangoldt-nonneg)}
      - (auto simp: floor-le-iff)
  - \text{moreover have } x / \text{real } d \leq 1 + \text{real-of-int } [x / \text{real } d] \text{ for } d \text{ by linarith}
  - \text{hence } r x \leq \text{sum-upto } (\lambda d. \text{mangoldt } d * 1) x \text{ unfolding } \text{sum-upto-altdef eq}
    - \text{r-def using } x
      - \text{by (intro sum-mono mult-mono mangoldt-nonneg)}
        - (auto simp: less-imp-le[OF frac-lt-1] algebra-simps)
      - \text{ultimately show } \text{norm } (r x) \leq 1 * \text{norm } (\psi x) \text{ by (simp add: } \psi\text{-def)}
    - auto
  - \text{also have } \psi \in O(\lambda x. x) \text{ by (fact } \psi\text{-bigo)}
  - \text{finally have } r : r \in O(\lambda x. x) \cdot

\[\text{define } r' \text{ where } r' = (\lambda x::\text{real}. \text{sum-upto ln } x - x * \text{ln } x)\]
\[\text{have } r'-\text{bigo: } r' \in O(\lambda x. x)\]
\[\text{using stirling-weak-bigo unfolding } r'-\text{def} \cdot\]
\[\text{have ln-fact: } \text{ln } (\text{fact } n) = (\sum d=1..n. \text{ln } d) \text{ for } n\]
\[\text{by (induction } n) \text{ (simp-all add: ln-mult)}\]
\[\text{hence } r': \text{sum-upto ln } n = n * \text{ln } n + r' n \text{ for } n :: \text{real}\]
\[\text{unfolding } r'-\text{def } \text{sum-upto-altdef by (auto intro: sum.cong)}\]
\[\text{have eventually } (\lambda n. \text{sum-upto } (\lambda d. \text{mangoldt } d / d) n - \text{ln } n = r' n / n + r n / n) \text{ at-top}\]
\[\text{using eventually-gt-at-top}\]
\[\text{proof eventually-elim}\]
\[\text{fix } x :: \text{real assume } x : x > 0\]
\[\text{have sum-upto ln x = sum-upto } (\lambda n. \text{mangoldt } n * (\text{nat } [x / n])) x\]
\[\text{unfolding sum-upto-ln-cone-sum-upto-mangoldt} ..\]
\[\text{also have } \ldots = \text{sum-upto } (\lambda d. \text{mangoldt } d * (x / d)) x - r x\]
\[\text{unfolding sum-upto-def by (simp add: } \text{algebra-simps sum-subtractf } r\text{-def sum-upto-def)}\]
\[\text{also have } \text{sum-upto } (\lambda d. \text{mangoldt } d * (x / d)) x = x * \text{sum-upto } (\lambda d. \text{mangoldt } d / d) x\]
\[\text{unfolding sum-upto-def by (subst sum-distrib-left) (simp add: field-simps)}\]
\[\text{finally have } x * \text{sum-upto } (\lambda d. \text{mangoldt } d / \text{real } d) x = r' x + r x + x * \text{ln } x\]
\[\text{by (simp add: } r' \text{ algebra-simps)}\]
\[\text{thus sum-upto } (\lambda d. \text{mangoldt } d / d) x - \text{ln } x = r' x / x + r x / x\]
\[\text{using } x \text{ by (simp add: field-simps)}\]
\[\text{qed}\]
\[\text{hence } (\lambda x. \text{sum-upto } (\lambda d. \text{mangoldt } d / d) x - \text{ln } x) \in \Theta(\lambda x. r' x / x + r x / x)\]

83
by (rule bigtheta-cong)
also have \((\lambda x. r' x / x + r x / x) \in O(\lambda-. 1)\)
by (intro sum-in-bigo) (insert r r'-bigo, auto simp: landau-divide-simps)
finally show thesis.

qed

Combining these two gives us Mertens’ first theorem.

**Theorem** mertens-bounded: \((\lambda x. \mathcal{M} x - \ln x) \in O(\lambda-. 1)\)
**proof** –
define f where \(f = \text{sum-upto} (\lambda d. \text{mangoldt} d / d)\)
have \((\lambda x. (f x - \ln x) - (f x - \mathcal{M} x)) \in O(\lambda-. 1)\)
using sum-upto-mangoldt-over-id-asymptotics
sum-upto-mangoldt-over-id-minus-phi-bounded
unfolding f-def by (rule sum-in-bigo)
thus thesis by simp

qed

**Lemma** primes-M-bigo: \(\mathcal{M} \in O(\lambda x. \ln x)\)
**proof** –
have \((\lambda x. \mathcal{M} x - \ln x) \in O(\lambda-. 1)\)
by (rule mertens-bounded)
also have \((\lambda x. \ln x) \in O(\lambda x. \ln x)\)
by real-asymp
finally have \((\lambda x. \mathcal{M} x - \ln x + \ln x) \in O(\lambda x. \ln x)\)
by (rule sum-in-bigo) auto
thus thesis by simp

qed

4 The Prime Number Theorem

**Theory** Prime-Number-Theorem
**Imports**
Newman-Ingham-Tauberian
Prime-Counting-Functions

**Begin**

4.1 Constructing Newman’s function

Starting from Mertens’ first theorem, i.e. \(\mathcal{M}(x) = \ln x + O(1)\), we now want to derive that \(\mathcal{M}(x) = \ln x + c + o(1)\). 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** \texttt{fds-newman where}
\[
\texttt{fds-newman} = \texttt{fds} (\lambda n. \texttt{complex-of-real} (\Re n))
\]

**lemma** \texttt{fds-nth-newman:}
\[
\texttt{fds-nth fds-newman n} = \texttt{of-real} (\Re n) \\
\textbf{by (simp add: fds-newman-def fds-nth-fds)}
\]

**lemma** \texttt{norm-fds-nth-newman:}
\[
\texttt{norm (fds-nth fds-newman n)} = \Re n \\
\textbf{by (intro abs-of-nonneg sum-nonneg divide-nonneg-pos (auto dest: prime-ge-1-nat))}
\]

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** \texttt{bounded-coeffs-newman-minus-deriv-zeta:}
\[
\texttt{defines f} \equiv \texttt{fds-newman + fds-deriv fds-zeta} \\
\texttt{shows Bseq (} \lambda n. \texttt{fds-nth f n) }
\]

**proof**
\[
\texttt{have (} \lambda n. \Re (\texttt{real n}) - \ln (\texttt{real n})\} \in O(\lambda^- 1) \\
\texttt{using mertens-bounded by (rule landau-o.big.compose) real-asynmp} \\
\texttt{from natfun-bigo-1E[OF this, of 1]} \\
\texttt{obtain c where c: c \geq 1} \land \forall n. \Re (\texttt{real n}) - \ln (\texttt{real n}) \leq c \texttt{by auto} \\
\texttt{show } ?\texttt{thesis c: c \geq 1} \land \forall n. \Re (\texttt{real n}) - \ln (\texttt{real n}) \leq c \texttt{by auto} \\
\texttt{qed (insert c, auto)} \\
\texttt{qed (insert c, auto)} \\
\texttt{qed}
\]

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** \texttt{abs-conv-abscissa-newman: abs-conv-abscissa fds-newman \leq 1}
\[
\texttt{and conv-abscissa-newman: conv-abscissa fds-newman \leq 1}
\]

**proof**
\[
\texttt{define f where f} = \texttt{fds-newman + fds-deriv fds-zeta} \\
\texttt{have abs-conv-abscissa f \leq 1} \\
\texttt{using bounded-coeffs-newman-minus-deriv-zeta unfolding f-def}
\]
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 eval-fds-newman-conv-infsetsum:**

- **assumes** $s$: $\Re s > 1$
- **shows** $\text{eval-fds} f s = (\sum_{a \in \text{prime}} (a \cdot \ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p s)$

- **proof**
  - **from** $s$ **have** $\text{conv}: \text{fds-}abs\text{-}converges f s$
  - **define** $f$ **where** $f = (\lambda n. \ln (\text{real } p) / \text{real } p / \text{of-nat } n \text{ powr } s)$
  - **have** $eq: (\sum_{a \in \{p..\}, f n p} = (\sum_{a \times \{p..\}} (\ln (\text{real } p) / \text{of-nat } p) * (1 / \text{of-nat } x \text{ powr } s))$
    - **by** (simp add: f-def)
    - **also have** $\ldots = (\ln (\text{real } p) / \text{of-nat } p) * (\sum_{a \times \{p..\}} 1 / \text{of-nat } x \text{ powr } s)$
  - using $\text{abs-summable-hurwitz-zeta}[\text{of s 0 p}]$ that $s$
  - **by** (intro infsetsum-cmult-right) (auto dest: prime-gt-0-nat)
  - **also have** $(\sum_{a \times \{p..\}}, 1 / \text{of-nat } x \text{ powr } s) = \text{hurwitz-zeta } p s$
  - using $s$ that **by** (subt hurwitz-zeta-conv-infsetsum(2))
    - (auto dest: prime-gt-0-nat simp: field-simps powr-minus)
  - **finally show** $\theta$thesis .

- **qed**
also have \( \vdash (\lambda(p, n), f n p) \) \( \text{abs-summable-on} \)
\( (\lambda(n, p), (p, n)) \) \( (\Sigma p: \text{prime } p \land p \leq n) \)
by (\( \text{subst } \text{abs-summable-on-reindex iff [symmetric]} \) (auto simp: case-prod-unfold inj-on-def)
also have \( (\lambda(n, p), (p, n)) \) \( (\Sigma p: \text{prime } p \land p \leq n) \) = \( (\Sigma p: \{p. \text{prime } p\}, \{p..\}) \) by auto
finally have \( \text{summable 2: } (\lambda(p, n), f n p) \) \( \text{abs-summable-on} \ldots \).
from \( \text{abs-summable-on-Sigma-project1[OF this]} \)
have \( (\lambda p, \sum a n \in \{p..\}, f n p) \) \( \text{abs-summable-on} \) \( \{p. \text{prime } p\} \) by auto
also have \( \?this \vdash (\lambda p. \ln (\text{real } p) / \text{real } p * \text{hurwitz-zeta } p s) \) \( \text{abs-summable-on} \)
\( \{p. \text{prime } p\} \)
by (intro \( \text{abs-summable-on-cong eq} \) auto)
finally show \( \ldots \).

have \( \text{eval-fds fds-newman } s = \)
\( (\Sigma a n, \sum p \mid \text{prime } p \land p \leq n. \ln (\text{real } p) / \text{real } p / \text{of-nat } n \text{ pow } s) \)
using \( \text{conv} \) by (simp add: \( \text{eval-fds-altdef fds-nth-newman sum-divide-distrib primes-M-def prime-sum-upto-def} \))
also have \( \ldots = (\Sigma a n. \sum a p \mid \text{prime } p \land p \leq n. f n p) \)
unfolding \( f-def \) by (\( \text{subst infsetsum-finite} \) auto)
also have \( \ldots = (\sum a (p, n) \in (\lambda(n, p), (p, n)) \) \( (\Sigma p: \text{prime } p \land p \leq n) \) \( f n p) \)
using \( \text{summable 1} \) by (\( \text{subst infsetsum-Sigma} \) auto)
also have \( \ldots = (\sum a (p, n) \in (\lambda(n, p), (p, n)) \) \( (\Sigma p: \text{prime } p \land p \leq n) \) \( f n p) \)
by (\( \text{subst infsetsum-reindex} \) (auto simp: case-prod-unfold inj-on-def)
also have \( (\lambda(n, p), (p, n)) \) \( (\Sigma p: \text{prime } p \land p \leq n) \) = \( (\Sigma p: \{p. \text{prime } p\}, \{p..\}) \) by auto
also have \( (\sum a (p, n) \in \ldots f n p) = (\sum a p \mid \text{prime } p. \sum a n \in \{p..\}, f n p) \)
using \( \text{summable 2} \) by (\( \text{subst infsetsum-Sigma} \) auto)
also have \( (\sum a p \mid \text{prime } p. \sum a n \in \{p..\}, f n p) = (\sum a p \mid \text{prime } p. \ln (\text{real } p) / \text{real } p * \text{hurwitz-zeta } p s) \)
by (intro \( \text{infsetsum-cong eq} \) auto)
finally show \( \text{eval-fds fds-newman } s = \)
\( (\sum a p \mid \text{prime } p. (\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p s) \).

\( \text{qed} \)

We now define a meromorphic continuation of \( f(s) \) on \( \Re(s) > \frac{1}{2} \).
To construct \( f(s) \), we express it as
\[
 f(s) = \frac{1}{s - 1} \left( \tilde{f}(s) - \frac{\zeta'(s)}{\zeta(s)} \right),
\]
where \( \tilde{f}(s) \) (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 Weierstraß M test. \( \frac{\zeta'(s)}{\zeta(s)} \) 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

fixes $A :: \text{nat} \Rightarrow \text{complex} \Rightarrow \text{complex}$ and $B :: \text{nat} \Rightarrow \text{complex} \Rightarrow \text{complex}$

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

defines $B \equiv (\lambda p s. \text{of-real (ln (real p))} / \text{of-nat p} * A p s)$

begin

definition pre-newman :: $\text{complex} \Rightarrow \text{complex}$ where

$\text{pre-newman s} = (\sum p. \text{if prime p then B p s else 0})$

definition newman where

$newman s = 1 / (s - 1) * (\text{pre-newman s} - \text{deriv zeta s} / \text{zeta s})$

The sum used in the definition of pre-newman converges uniformly on any disc within the half-space with $\Re(s) > 1/2$ by the Weierstraß M test.

lemma uniform-limit-pre-newman:

assumes $r :: r \geq 0 \text{ Re s} - r > 1/2$

shows uniform-limit($cball s r$)($\lambda n s. \sum p < n. \text{if prime p then B p s else 0}$)

begin

show thesis unfolding pre-newman-def

proof (intro Weierstrass-m-test-ev[OF eventually-mono[OF eventually-gt-at-top[of I]]]) ballI)

show summable $M$

proof (rule summable-comparison-test-bigo)

define $\varepsilon$ where $\varepsilon = \text{min} (2 * x - 1) x / 2$

from $\frac{x}{2} > 1 / 2$, have $\varepsilon : \varepsilon > 0 1 + \varepsilon < 2 * x 1 + \varepsilon < x + 1$

by (auto simp: $\varepsilon$-def min-def field-simps)

show $M \in O(\lambda n. n \text{ powr } (-1 - \varepsilon))$ unfolding M-def distrib-left

by (intro sum-in-bigo) (use $\varepsilon$ in real-asympt)+

from $\varepsilon$ show summable $(\lambda n. n \text{ norm } (n \text{ powr } (-1 - \varepsilon)))$

by (simp add: summable-real-powr-iff)

qed

next

88
fix \( p : \text{nat} \) and \( z \) assume \( p : p > 1 \) and \( z : z \in \mathbb{C} \) and \( r \)

from \( z r \) \( \Re[\text{of } z] \) have \( x : \Re z \geq x \) and \( \Re z > 1 / 2 \) and \( \Re z > 1 / 2 \)

using abs-\( \Re \)-le-cmod[of \( s - z \)] by (auto simp: x-def algebra-simps dist-norm)

have norm-z: norm \( z \leq \) norm \( s + r \)

using \( z \text{ norm-triangle-ineq2[of } z s \) r \) by (auto simp: dist-norm norm-minus-commute)

from \( p > 1 \) and \( x \) and \( r \) have \( M p \geq 0 \)

by (auto simp: C-def M-def intro: mult-nonneg-nonneg add-nonneg-nonneg divide-nonneg-pos)

have bound: norm \((z - 1) \ast \text{pre-zeta } p \) \( z \) \( \leq \)

\( \text{norm } (z - 1) \ast (\text{norm } z / (\Re z \ast p \text{ powr } \Re z)) \)

using pre-zeta-bound[of \( p \) \( p \ast \Re z > 1 / 2 \)]

unfolding norm-mult by (intro mult-mono pre-zeta-bound) auto

have norm \((B p z) = \ln p / p \ast \text{norm } (A p z) \)

unfolding B-def using \( p > 1 \) by (simp add: B-def norm-mult norm-divide)

also have \dots \( \leq \) \( \ln p / p \ast (\text{norm } (z - 1) \ast \text{norm } z / \Re z / p \text{ powr } \Re z + p / (p \text{ powr } \Re z \ast (p \text{ powr } \Re z - 1))) \)

unfolding A-def using \( p > 1 \) and \( \Re z > 1 / 2 \) and \( \text{bound} \)

by (intro mult-left-mono order.trans[OF norm-triangle-ineq4 add-mono] mult-left-mono)

(auto simp: norm-mult norm-divide mult-nonneg norm-powr-real-powr intro!: divide-left-mono order.trans[OF norm-triangle-ineq2])

also have \dots \( = \) \( \ln p \ast (\text{norm } (z - 1) \ast \text{norm } z / \Re z / p \text{ powr } (\Re z + 1) \)

\( + \)

\( 1 / (p \text{ powr } \Re z \ast (p \text{ powr } \Re z - 1))) \)

using \( p > 1 \) by (simp add: field-simps powr-add powr-minus)

also have norm \((z - 1) \ast \text{norm } z / \Re z / p \text{ powr } (\Re z + 1) \leq C / p \text{ powr } (x + 1) \)

unfolding C-def using \( r \ast \Re z > 1 / 2 \) \( \text{norm-z } p \) \( x \)

by (intro mult-mono frac-le powr-powr order.trans[OF norm-triangle-ineq4])

auto

also have \( 1 / (p \text{ powr } \Re z \ast (p \text{ powr } \Re z - 1)) \leq \)

\( 1 / (p \text{ powr } x \ast (p \text{ powr } x - 1))) \) using \( p > 1 \) \( x \)

by (intro divide-left-mono mult-mono powr-powr mult-pos-pos)

(auto simp: ge-one-powr-ge-zero)

finally have norm \((B p z) \leq M p \)

using \( p > 1 \) by (simp add: mult-left-mono M-def)

with \( M p \geq 0 \) show norm \((\text{if prime } p \text{ then } B p z \text{ else } 0) \leq M p \) by simp

qed

lemma sums-pre-newman: \( \text{Re } s > 1 / 2 \implies (\lambda p. \text{if prime } p \text{ then } B p s \text{ else } 0) \)

sums pre-newman s

using tends-to-uniform-limitI[OF uniform-limit-pre-newman[of \( 0 \) s]] by (auto simp: sums-def)

lemma analytic-pre-newman [THEN analytic-on-subset, analytic-intros]:

pre-newman analytic-on \{s. \text{Re } s > 1 / 2\}

proof –

89
have holoc : (λs:complex. if prime p then B p s else 0) holomorphic-on X
if X ⊆ {s. Re s > 1 / 2} for X and p :: nat using that
by (cases prime p)
(auto intro: holomorphic-intros simp: B-def A-def dest: prime-gt-1-nat)
have holo': pre-newman holomorphic-on ball s r if r ≥ 0 Re s - r > 1 / 2
for s r
proof
from r have Re: Re z > 1 / 2 if dist s z ≤ r for z
using abs-RC mod[of s - z] r 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]])
(insert that Re, auto intro!)
holomorphic-intros holo
qed

lemma holomorphic-pre-newman [holomorphic-intros]:
X ⊆ {s. Re s > 1 / 2} ⇒ pre-newman holomorphic-on X
using analytic-pre-newman by (rule analytic-imp-holomorphic)

lemma eval-fds-newman:
assumes s : Re s > 1
shows eval-fds fds-newman s = newman s
proof
have eq: (ln (real p) / real p) * hurwitz-zeta p s =
1 / (s - 1) * (ln (real p) / (p powr s - 1) + B p s)
if p: prime p for p
proof
have (ln (real p) / real p) * hurwitz-zeta p s =
ln (real p) / real p * (p powr (1 - s) / (s - 1) + pre-zeta p s)
using s by (auto simp add: hurwitz-zeta-def)
also have ... = 1 / (s - 1) * (ln (real p) / (p powr s - 1) + B p s)
using p s by (simp add: divide-simps powr-diff B-def)
(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}
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** $0 < \Re s \neq 1$  
**Shows** $\text{deriv} \ zeta \ s = \frac{(\text{pre-zeta } 1 \ z \ast (z - 1)) \ s \ast (s - 1)^2}{((s - 1) \ast (s - 1)) \ s \ast (s - 1) + \text{pre-zeta } 1 \ s}$  

**Proof** (rule DERIV-imp-deriv)  
**Have** [derivative-intros]: $(\text{pre-zeta } 1 \ has-field-derivative \ deriv \ (\text{pre-zeta } 1) \ s)$ (at s)  

by (intro holomorphic-deriv[of - UNIV] holomorphic-intros) auto  

have $\ast$: $\text{deriv} (\lambda z. \text{pre-zeta } 1 \ z \ast (z - 1)) \ s \ast (s - 1) + \text{pre-zeta } 1 \ s$  

by (subst deriv-mult  

(auto intro: holomorphic-on-imp-differentiable-at[of - UNIV] holomorphic-intros)

**Hence** $((\lambda s. \text{pre-zeta } 1 \ s \ast 1 \ / \ (s - 1)) \ has-field-derivative \ deriv \ (\text{pre-zeta } 1) \ s \ast 1 \ / \ ((s - 1) \ast (s - 1)))$ (at s)  

using assms by (auto intro: derivative-eq-intros)

also have $\text{deriv} \ (\text{pre-zeta } 1) \ s \ast 1 \ / \ ((s - 1) \ast (s - 1)) = \ ?\text{rhs}$
using \ast\ assms by (simp add: divide-simps power2-eq-square, simp add: field-simps)
also have \((\text{As. pre-zeta } 1 s + 1 / (s - 1))\) has-field-derivative \(\text{?rhs}\) (at \(s\)) \(\longleftarrow\) \(\text{zeta}\) has-field-derivative \(\text{?rhs}\) (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 \((s - 1)\zeta'(s) - \zeta'(s)/\zeta(s)\) is analytic for \(\Re(s) \geq 1\):

lemma analytic-zeta-derivdiff:
obtains a where
\((\lambda z. \text{if } z = 1 \text{ then } a \text{ else } (z - 1) * \text{ deriv zeta } z - \text{ deriv zeta } z / \text{ zeta } z)\)
analytic-on \(\{s, \Re s \geq 1\}\)

proof
have neq: \(\text{pre-zeta } 1 z * (z - 1) + 1 \neq 0\) 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 \(?g = \lambda z. (1 - inverse (\text{pre-zeta } 1 z * (z - 1) + 1)) * ((z - 1) * \text{ deriv } ((\lambda a. \text{pre-zeta } 1 a * (a - 1))) z) - (\text{pre-zeta } 1 z * (z - 1) + 1)\)

show \((\lambda z. \text{if } z = 1 \text{ then } \text{ deriv } ?g 1 \text{ else } (z - 1) * \text{ deriv zeta } z - \text{ deriv zeta } z / \text{ zeta } z)\) finally show \(\text{ analytic-on } \{s, \Re s \geq 1\}\)

next

show \(\exists d>0. \forall w\in \text{ball } z d - \{1\}. \exists w = (w - 1) * \text{ if } w\)
if \(z: z \in \{s, 1 \leq \Re s\}\) for \(z\)

proof

have \(*: \text{isCont } (\lambda z. \text{pre-zeta } 1 z * (z - 1) + 1) z\)
by (auto intro!: continuous-intros)

obtain e where \(e > 0\) and \(e: \forall y. \text{dist } y < e \Longrightarrow \text{pre-zeta } (\text{Suc } 0) y * (y - 1) + 1 \neq 0\)
using continuous-at-avoid [OF \(*: neq\of z\)] \(\text{z by auto}\)

show \(?\text{thesis}\)

proof (intro exI ballI conjI)

fix \(w\)
assume \(w: w \in \text{ball } z (\min e 1) - \{1\}\)
then have \(\Re w > 0\)
using complex-Re-le-cmod [of \(z - w\)] \(\text{z by (simp add: dist-norm)}\)

with \(w\) show \(?g w = (w - 1) * (\text{if } w = 1 \text{ then } \text{deriv } ?g 1 \text{ else}\)

\((w - 1) * \text{ deriv zeta } w - \text{ deriv zeta } w / \text{ zeta } w)\)

by (subst (1 2) \(\text{deriv-zeta-eq}\),
\(\text{simp-all add: zeta-def hurwitz-zeta-def divide-simps e power2-eq-square})
\(\text{(simp-all add: algebra-simps)}\)?)

qed (use \(\{e > 0\}\) in auto)

qed

qed auto
Finally, \( f(s) + \zeta'(s) + c\zeta(s) \) is analytic.

**lemma analytic-newman-variant:**

**obtains** \( c \) \( a \) where

\[
(\lambda z. \; \text{if } z = 1 \; \text{then } a \; \text{else } \text{newman } z + \text{deriv } \zeta \; z + c \times \zeta \; z) \quad \text{analytic-on} \quad \{ s. \; \text{Re } s \geq 1 \}
\]

**proof** –

**obtain** \( c \) \( a \) where

\[
(\lambda z. \; \text{if } z = 1 \; \text{then } c \; \text{else } \text{deriv } \zeta \; z - \text{deriv } \zeta \; z / \zeta \; z) \quad \text{analytic-on} \quad \{ s. \; \text{Re } s \geq 1 \}
\]

**using** analytic-zeta-derivdiff \( \text{by } \) blast

**let** \( ?g = \lambda z. \text{pre-newman } z + 
\]

\[
(\text{if } z = 1 \; \text{then } c \; \text{else } \text{deriv } \zeta \; z - \text{deriv } \zeta \; z / \zeta \; z) - (c + \text{pre-newman } 1) \times (\text{pre-zeta } 1 \; z + (z - 1) + 1)
\]

**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)) \times \zeta \; z) \quad \text{analytic-on} \quad \{ s. \; \text{Re } s \geq 1 \} \quad \text{(is } ?f \text{ analytic-on -)}
\]

**proof** \( \text{(rule pole-theorem-analytic-0)} \)

**show** \( ?g \) analytic-on \( \{ s. \; 1 \leq \text{Re } s \} \)

**by** \( \text{(intro } c \text{ analytic-intros) auto} \)

**next**

**show** \( \exists d > 0. \; \forall w \in \text{ball } d - \{ 1 \}. \; ?g \; w = (w - 1) \times ?f \; w \)

**if** \( z \in \{ s. \; 1 \leq \text{Re } s \} \) \( \text{for } z \) \( \text{using that} \)

**by** \( \text{(intro exI[of - 1], simp-all add: newman-def divide-simps zeta-def hurwitz-zeta-def)} \)

**auto simp: field-simps)?**

**qed**

**auto**

**with that show** \( ?thesis \) \( \text{by blast} \)

**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** \( c :: \text{real} \) \( a \) where

\[
(\lambda n. \; (\mathcal{M} \; n - \ln n + c) / n)
\]

**proof** –

**from** analytic-newman-variant \( \text{obtain } c \) \( a \) \( \text{where} \)

93
analytic: (λz. if z = 1 then a else newman z + deriv zeta z + c * zeta z)
  analytic-on {s. Re s ≥ 1} .

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-abscissa-add-leI conv-abscissa-deriv-le conv-abscissa-newman conv-abscissa-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-1)
    have (λn. ℝ (real n) - ln (real n)) ∈ O(λn. 1)
      using mertens-bounded by (rule landau-o.big.compose) real-asymptotic
    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-Real sphere)
      qed (insert c', auto intro: add-pos-nonneg)
    thus fds-nth F ∈ O(λn. 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)
    by (intro summable-cong eventually-mono[of eventually-gt-at-top[of 0]])
      (auto simp: F-def fds-nth-newman fds-nth-deriv fds-nth-zeta scaleR-conv-of-real intro!: sum.cong dest: prime-gt-0-nat)
  finally have summable (λn. (ℝ n - Re (Ln (of-nat n)) + Re c) / n)
    by (auto dest: summable-Real)
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 : \{ \sum i = M \cdot N. \ a i / i < d \text{ and } \forall n. \ a n + \ln n \leq a (\text{Suc} n) + \ln (\text{Suc} n) \}

and $0 < M M < N$

**shows** $a M \leq d * N / (\text{real} N - \text{real} M) + (\text{real} N - \text{real} M) / M \wedge -a N \leq d * N / (\text{real} N - \text{real} M) + (\text{real} N - \text{real} M) / M$

**proof**

have $0 \leq d$

using assms by linarith+

then have $0 \leq d * N / (N - M + 1)$ by simp

then have 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 linarith

have le-a-ln: $a m + \ln m \leq a n + \ln n$ if $n \geq m$ for $m n$

by (rule transitive-stepwise-le) (use le that in auto)

have $*: x \leq b \wedge y \leq b$ if $a \leq b x \leq a y \leq a$ for $a b x y :: \text{real}$

using that by linarith

**show** $?thesis$

**proof** (rule *)

**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 $\langle 0 < M, \ M < N, \ \ln \text{-le-minus-one} \ [\text{of} \ N / M \rangle$

by (simp add: of-nat-diff) (simp add: divide-simps)

**next**

have $a M - \ln (N / M) \leq (d * N) / (N - M + 1)$

**proof** (rule le-dN)

assume $0: 0 \leq a M - \ln (N / M)$

have $(\text{Suc} N - M) * (a M - \ln (N / M)) / N = (\sum i = M \cdot N. \ (a M - \ln (N / M))) / N$

by simp

also have $\ldots \leq (\sum i = M \cdot N. \ a i / i)$

**proof** (rule sum_mono)

**fix** $i$

assume $i: i \in \{M \cdot N\}$

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-eg 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 \((a \cdot M - \ln (N / M)) / i \leq a / i \cdot \ln(i)
\)
using `assms` by \(\text{intro divide-right-mono}\) \(\text{auto simp: ln-div field-simps}\)
finally show \((a \cdot M - \ln (N / M)) / \text{real} \cdot N \leq a / i \cdot \text{real} \cdot i\).
Qed
finally have \((\text{Suc } N) - M) * (a \cdot M - \ln (N / M)) / N \leq \sum i = M..N. a
\)
\(i / i\)
by `simp`
also have \(\ldots \leq d\) using \(d\) by `simp`
finally have \((\text{Suc } N) - M) * (a \cdot M - \ln (N / M)) / N \leq d\).
then show `thesis`
using \((M < N)\) by \(\text{simp add: of-nat-diff field-simps}\)
Qed
also have \(\ldots \leq d \cdot N / (N - M)\)
using `assms(1,4)` by \(\text{simp add: field-simps}\)
finally show \(a \cdot M \leq d \cdot N / (N - M) + \ln (N / M)\) by `simp`
Next
have \(- a \cdot N - \ln (N / M) \leq (d \cdot N) / (N - M + 1)\)
proof \(\text{rule le-dN}\)
assume \(0: 0 \leq - a \cdot N - \ln (N / M)\)
have \((\sum i = M..N. a / i) \leq (\sum i = M..N. (a \cdot N + \ln (N / M)) / N)\)
proof \(\text{rule sum-mono}\)
fix \(i\)
assume \(i: i \in \{M..N\}\)
with \(\langle 0 < M \rangle\) have \(0 < i\) by `auto`
have \(a / i + \ln (\text{real} \cdot M) \leq a / i + \ln (\text{real} \cdot N)\)
by \(\text{rule order.trans[OF - a-ln[of i N]]}\) \(\text{use i assms in auto}\)
hence \(a / i / i \leq (a / i + \ln (N / M)) / i\)
using `assms(3,4)` by \(\text{intro divide-right-mono}\) \(\text{auto simp: field-simps}\)
\(\text{ln-die}\)
also have \(\ldots \leq (a / i + \ln (N / M)) / N\)
using \(\langle i \cdot i > 0\rangle\) \(\langle \text{intro divide-left-mono-neg} \rangle\) `auto`
finally show \(a / i / i \leq (a / i + \ln (N / M)) / N\).
Qed
also have \(\ldots = ((\text{Suc } N) - M) * (a / (N + \ln (N / M))) / N\)
by `simp`
finally have \((\sum i = M..N. a / i) \leq (\text{real} (\text{Suc } N) - \real M) * (a / (N + \ln (N / M))) / N\)
using \((M < N)\) by \(\text{simp add: of-nat-diff}\)
then have 
\(-((\text{real} (\text{Suc } N) - \real M) * (a / (N + \ln (N / M))) / N) \leq \sum i\)
\(= M..N. a / i / i\)
by `linarith`
also have \(\ldots \leq d\) using \(d\) by `simp`
finally have 
\(-((\text{real} (\text{Suc } N) - \real M) * (a / (N + \ln (N / M))) / N) \leq d\).
then show `thesis`
using \((M < N)\) by \(\text{simp add: of-nat-diff field-simps}\)
Qed
also have \(\ldots \leq d \cdot N / \text{real} (N - M)\)
using \(\langle 0 < M \rangle\) \(\langle M < N \rangle\) \(\langle 0 \leq d \rangle\) by \(\text{simp add: field-simps}\)
finally show \(- a \cdot N \leq d \cdot N / \text{real} (N - M) + \ln (N / M)\) by `simp`
proposition sum-goestozero-theorem:
assumes summ: summable (λi. a i / i)
and le: \( \forall n. a n + \ln n \leq a (Suc n) + \ln (Suc n) \)
shows a \( \longrightarrow 0 \)
proof (clarsimp simp: lim-sequentially)
fix r :: real
assume r > 0
have \(*\): \( \exists n_0. \forall n \geq n_0. |a n| < \varepsilon \) if \( \varepsilon > 0 \)
proof –
have 0 < \((\varepsilon / 8)^2\) using (0 < \varepsilon) by simp
then obtain N0 where N0: \( \forall m n. m \geq N0 \implies \norm{\sum_{k=m..n} (\lambda i. a i / i)} < (\varepsilon / 8)^2 \)
by (metis summable-partial-sum-bound summ
obtain N1 where real N1 > 4 / \varepsilon by auto
hence N1 ≠ 0 and N1: 1 / real N1 < \varepsilon / 4 using \varepsilon
by (auto simp: divide-simps mult-ac intro: Nat.gr0I)
have |a n| < \varepsilon if n ≥ 2 \ast N0 + N1 + 7 for n
proof –
define k where k = \lfloor n * \varepsilon / 4 \rfloor
have n * \varepsilon / 4 > 1 and n * \varepsilon / 4 ≤ n / 4 and n / 4 < n
using less-le-trans[OF N1, of n / N1 * \varepsilon / 4] \varepsilon by auto
hence k: k > 0 \ast k ≤ n nat k < n (n * \varepsilon / 4) - 1 < k k ≤ (n * \varepsilon / 4)
unfolding k-def by linarith
have \(-a n < \varepsilon\)
proof –
have N0 ≤ n - nat k
using n k by linarith
then have \(*\): \( \sum_{k=n-nat k} a k / k < (\varepsilon / 8)^2 \)
using N0 [of n - nat k n] by simp
have \(-a n ≤ (\varepsilon / 8)^2 \ast n / (n * \varepsilon / 4) + n * \varepsilon / 4 / (n - k)\)
using sum-goestozero-lemma [OF * le, THEN conjunct2] k by (simp add: of-nat-diff k-def)
also have \(\ldots < \varepsilon\)
proof –
have \(\varepsilon / 16 \ast n / k < 2\)
using k by (auto simp: field-simps)
then have \(\varepsilon \ast (\varepsilon / 16 \ast n / k) < \varepsilon \ast 2\)
using \(\varepsilon\) mult-less-cancel-left-pos by blast
then have \((\varepsilon / 8)^2 \ast n / k < \varepsilon / 2\)
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 \(k \cdot \varepsilon\) by simp
also have \(\ldots \leq \varepsilon \cdot \text{real } n\) using \(k\) by (auto simp: field-simps)
finally show \(?thesis\) using \(k\) by (auto simp: field-simps)
qed
ultimately show \(?thesis\) unfolding \(k\)-def by linarith
qed
finally show \(?thesis\).
qed
moreover have \(a \cdot n < \varepsilon\)
proof
- have \(N_0 \leq n\) using \(n, k\) by linarith
then have \(\sum k = n \cdot n + \text{nat } k \cdot a \cdot k / k < (\varepsilon / 8)^2\)
  using \(N_0\) [of \(n, n + \text{nat } k\)] by simp
have \(a \cdot n \leq (\varepsilon / 8)^2 \cdot (n + \text{nat } k) / k + n\)
  using sum-goestozero-lemma [OF \(\ast\), THEN conjunct1]\(k\) by (simp add:
of-nat-diff)
also have \(\ldots < \varepsilon\)
proof
- have \(4 \leq 28 \cdot \text{real-of-int } k\) 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 add: mult.commute \(k\)-def)
    then have \(\varepsilon \cdot n < 4 \cdot k + 4\) by (simp add: divide-simps)
    also have \(\ldots \leq 8 \cdot k\) using \(k\) by auto
    finally have \(1: \varepsilon \cdot \text{real } n < 8 \cdot k\).
    have \(2: \varepsilon \cdot k < k\) using \(k \cdot \varepsilon\) by simp
    show \(?thesis\) using \(k\) add-strict-mono [OF \(1, 2\)] by (simp add: algebra-simps)
  qed
ultimately show \(?thesis\) by linarith
qed
finally show \(?thesis\).
qed
ultimately show \(?thesis\) by force
qed
then show \(?thesis\) by blast
qed
show \(\exists n_0. \forall n \geq n_0. \|a 
using \(*\) [of min \(r (1/5)\)] \langle 0 < r \rangle\) by force
qed

This leads us to the main intermediate result:

\textbf{lemma} \text{Mertens-convergent: }\text{convergent } (\lambda n::\text{nat} . \Re \cdot n - \ln n)

\textbf{proof} –
obtain \( c \) where \( c \) is summable \((\lambda n. (\mathfrak{M} n - \ln n + c) / n)\)
by (blast intro: mertens-summable)
then obtain \( l \) where \( l \) is \( \lambda n. (\mathfrak{M} n - \ln n + c) / n \) sums \( l \)
by (auto simp: summable-def)
have \( \ast : \lambda n. (\mathfrak{M} n - \ln n + c) \rightarrow 0 \)
by (rule sum-goestozero-theorem[OF \( c \)])
auto
hence \( \lambda n. (\mathfrak{M} n - \ln n) \rightarrow -c \)
by (simp add: tendsto-iff dist-norm)
thus \(?\)thesis by (rule convergentI)
qed

Corollary \( \mathfrak{M} \)-minus-ln-limit:
obtains \( c \) where \( \lambda x :: \text{real}. \mathfrak{M} x - \ln x \rightarrow c \) at-top
proof
from Mertens-convergent obtain \( c \) where \( \lambda n. (\mathfrak{M} n - \ln n) \rightarrow c \)
by (auto simp: convergent-def)
hence \( 1 : \lambda x :: \text{real}. (\text{nat} \lfloor x \rfloor) - \ln \lfloor x \rfloor \rightarrow c \) at-top
by (rule filterlim-compose real-asym)
have \( 2 : \lambda x :: \text{real}. \ln \lfloor x \rfloor - \ln x \rightarrow 0 \) at-top
by (simp add: tendsto-iff dist-norm)
have \( 3 : \lambda x. (\mathfrak{M} x - \ln x) \rightarrow c \) at-top
using tendsto-add[OF 1 2] by simp
with that show \(?\)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 \( \mathfrak{M} \) function, is \( \vartheta(x) \sim x \).

We know that \( \mathfrak{M}(x) \) has the asymptotic expansion \( \mathfrak{M}(x) = \ln x + c + o(1) \). We also know that

\[ \vartheta(x) = x\mathfrak{M}(x) - \int_2^x \mathfrak{M}(t) \, dt. \]

Substituting in the above asymptotic equation, we obtain:

\[
\begin{align*}
\vartheta(x) &= x\ln x + cx + o(x) - \int_2^x \ln t + c + o(1) \, dt \\
&= x\ln x + cx + o(x) - (x\ln x - x + cx + o(x)) \\
&= x + o(x)
\end{align*}
\]
In conclusion, \( \vartheta(x) \sim x \).

**Theorem** \( \vartheta \)-asymptotics: \( \vartheta \sim [\text{at-top}] (\lambda x. \ x) \)

**Proof**

- From \( \mathcal{M} \text{-minus-ln-limit} \) obtain \( c \) where \( c \) : \((\lambda x. \ \mathcal{M} \ x - \ln x) \longrightarrow c \) \( \text{at-top} \) by auto
- Define \( r \) where \( r = (\lambda x. \ \mathcal{M} \ x - \ln x - c) \)
- Have \( \mathcal{M} \text{-expand}: \mathcal{M} = (\lambda x. \ \ln x + c + r \ x) \) by (simp add: \( r \)-def)
- Have \( r: r \in o(\lambda\cdot) \) unfolding \( r \)-def using \( \text{tendsto-add}[\text{OF} c \ \text{tendsto-const}[o \ (-c)]] \) by (intro \( \text{smalloI-tendsto} \) 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 \( \text{unfolding} \) \( r \)-def by (intro \( \text{integrable-diff} \) \( \text{integrable-primes-M} \) (auto intro: \( \text{integrable-continuous-real} \) \( \text{continuous-intros} \))
  - Hence \( \text{integral}: (r \text{ has-integral} r' \ x) \ \{2..x\} \) if \( x \geq 2 \) for \( x \) by (auto simp: \( \text{has-integral-if} \) \( r' \)-def)
- Have \( r': r' \in o(\lambda x. \ x) \) using \( \text{integrable-r unfolding} \) \( r' \)-def
  - By (intro \( \text{integral-smallo}[\text{OF} r] \) (auto simp: \( \text{filterlim-ident} \))

**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 (intro \( \text{asympt-equiv-refl-eq} \) \( \text{eventually-mono}[\text{OF} \ \text{eventually-gt-at-top}] \))
  - Fix \( x :: \text{real} \) assume \( x: x > 2 \)
  - Have \( \mathcal{M} \ \text{has-integral} ((x \ast \ln x - x + c \ast x) - (2 \ast \ln 2 - 2 + c \ast 2) + r') \)
    - By (auto simp: \( \text{field-simps} \) \( \mathcal{M} \text{-expand} \) \( C \)-def)

**QED**

Also have \( (\lambda x. \ r \ x \ast x + C - r' \ x) \in o(\lambda x. \ x) \)

- Proof (intro \( \text{sum-in-smallo} \) \( r \) using \( \text{x} \)) (auto)

**QED** (insert \( \text{landau-o-small-big-mult}[\text{OF} r, \ of \ \lambda x. \ x] \) \( r' \), \( \text{simp-all} \))

- Hence \( (\lambda x. \ x + (r \ x \ast x + C - r' \ x)) \sim [\text{at-top}] (\lambda x. \ x) \)
  - Finally show \( \text{thesis by auto} \)

**QED**

The various other forms of the Prime Number Theorem follow as simple corollaries.

**Corollary** \( \psi \)-asymptotics: \( \psi \sim [\text{at-top}] (\lambda x. \ x) \)

- Using \( \psi \)-asymptotics \( \text{PNT4-imp-PNT5} \) by \( \text{simp} \)
corollary prime-number-theorem: \( \pi \sim [\lambda x. x / \ln x] \) using \( \theta \)-asymptotics PNT4-imp-PNT1 by simp

corollary ln-\( \pi \)-asymptotics: \( (\lambda x. \ln (\pi x)) \sim [\lambda x. \ln x] \) using \( \pi \)-asymptotics PNT1-imp-PNT1 by simp

corollary \( \pi \)-ln-\( \pi \)-asymptotics: \( (\lambda x. \pi x / \ln x) \sim [\lambda x. x] \) using \( \pi \)-ln-\( \pi \)-asymptotics PNT2-imp-PNT3 by simp

corollary nth-prime-asymptotics: \( (\lambda n. \pi^\prime n) \sim [\lambda n. \ln (\pi n)] \) using \( \pi \)-ln-\( \pi \)-asymptotics PNT2-imp-PNT3 by simp

The following versions use a little less notation.

corollary prime-number-theorem': \( (\lambda x. \pi x / (x / \ln x)) \longrightarrow 1) \) at-top using prime-number-theorem by (rule asymp-equivD-strong)[OF - eventually-mono[OF eventually-gt-at-top[of 1]]] auto

corollary prime-number-theorem'': \( (\lambda x. \text{card} \{ p. \text{prime} p \land \text{real} p \leq x \}) \sim [\lambda x. x / \ln x] \) proof - have \( \pi = (\lambda x. \text{card} \{ p. \text{prime} p \land \text{real} p \leq x \}) \) by (intro ext) (simp add: \( \pi \)-def prime-sum-upto-def) with prime-number-theorem show ?thesis by simp qed

corollary prime-number-theorem''': \( (\lambda n. \text{card} \{ p. \text{prime} p \land \text{real} p \leq n \}) \sim [\lambda n. \text{real} n / \ln (\text{real} n)] \) proof - have \( (\lambda n. \text{card} \{ p. \text{prime} p \land \text{real} p \leq \text{real} n \}) \sim [\lambda n. \text{real} n / \ln (\text{real} n)] \) using prime-number-theorem''' by (rule asymp-equiv-compose) (simp add: filterlim-real-sequentially) thus ?thesis by simp qed

corollary prime-number-theorem'''' (\lambda n. \text{card} \{ p. \text{prime} p \land \text{real} p \leq n \}) \sim [\lambda n. \text{real} n / \ln (\text{real} n)]

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. \( \zeta(n) = \ln n + O(1) \). 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 \( \zeta(n) - \ln n \in (-1 - \frac{9}{\pi^2}; \ln 4] \approx (-1.9119; 1.3863) \) 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** finite \( \{ i. \text{real} (p ^ (k * i + l)) \leq x \} \)

**proof** (rule finite-subset)

**show** \( \{ i. \text{real} (p ^ (k * i + l)) \leq x \} \subseteq \{ \text{nat} \mid x \} \)

**proof** safe

fix \( i \)
assume \( i : \text{real} (p ^ (k * i + l)) \leq x \)

have \( i < 2 ^ i \) by (rule less-exp)

also from assms have \( i \leq k * i + l \) by (cases \( k \))

hence \( 2 ^ i \leq (2 ^ (k * i + l)) :: \text{nat} \)

using assms by (intro power-increasing)

also have \( \ldots \leq p ^ (k * i + l) \) using assms by (intro power-mono)

also have \( \text{real} \ldots \leq x \) using \( i \) by simp

finally show \( i \leq \text{nat} \mid x \) 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 \( ((\lambda x:\text{real. ln} (x + 3) * (x + 3) \text{ powr} -2) \text{ has-integral} (\text{ln 3 + 1) / 3}) \)

(interior \( \{0..\} \))

using \( \text{ln-powr-has-integral-at-top[of 1 0 3 -2]} \)

by (simp add: interior-real-atLeast powr-minus)

hence \( ((\lambda x:\text{real. ln} (x + 3) * (x + 3) \text{ powr} -2) \text{ has-integral} (\text{ln 3 + 1) / 3}) \)

\( \{0..\} \)

by (subst (asm) has-integral-interior)

also have \( \text{this} \leftrightarrow ((\lambda x:\text{real. ln} (x + 3) / (x + 3) \text{ powr} 2) \text{ has-integral} (\text{ln 3 + 1) / 3}) \) \( \{0..\} \)
by (intro has-integral-cong) (auto simp: powr-minus field-simps)
finally have int: ....
have \( \exp (1/2 : real) \leq 2 \leq 2^2 \)
  using \( \text{exp-le} \) by (subst \( \exp-double \) [symmetric]) simp-all
hence \( \exp-half \): \( \exp (1/2 : real) \leq 2 \)
  by (rule power2-le-imp-le) auto

have mono: \( \ln x / x \leq \ln y / y \leq 2 \) if \( y \geq \exp(1/2) \) x \( y \) for \( x y : real \)
proof (rule DERIV-nonpos-imp-nonincreasing[of - - \( \lambda x. \ln x / x ^ 2 \)])
  fix t assume \( t \geq y t \leq x \)
  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 \( t y > 0 \) have \((\lambda x. \ln x / x ^ 2) \) has-field-derivative \((1 - 2 * \ln t) / t \leq 3)\) (at \( t \))
  by (auto intro!: derivative-eq-intros simp: eval-nat-numeral field-simps)
  moreover have \((1 - 2 * \ln t) / t \leq 3) \leq 0 \)
  using \( t \) that \( y > 0 \) \cdot \( \ln t \geq 1/2 \) by (intro divide-nonpos-pos) auto
  ultimately show \( 3 f'. ((\lambda x. \ln x / x ^ 2) \) has-field-derivative \( f' \) \)(at \( t \)) \wedge f' \leq 0 \) by blast
qed fact+

have \( \text{fds-converges} (\text{fds-deriv fds-zeta}) (2 : \text{complex}) \)
  by (intro fds-converges-derivative) auto
hence \((\lambda n. \text{of-real} (- \ln (\text{real} (\text{Suc} n)) / (\text{of-real} (\text{Suc} n)) \leq 2)) \) 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-realSuc)
  note * = \( \text{sums-split-initial-segment}[OF \text{sums-minus}[\text{OF \text{sums-Re}[OF this]], of \{3}]] \)
  have \((\lambda n. \ln (\text{real} (n+4)) / \text{real} (n+4) \leq 2) \) sums \((\text{Re} (\text{deriv zeta 2}) \leq (\ln 2 / 4 + \ln 3 / 9)) \)
  using * by (simp add: eval-nat-numeral)
  hence \((\text{Re} (\text{deriv zeta 2}) \leq (\ln 2 / 4 + \ln 3 / 9) = \)
    \((\sum n. \ln (\text{real} (\text{Suc} n) + 3) / (\text{real} (\text{Suc} n) + 3) \leq 2)) \)
  by (simp-all add: \text{sums iff algebra-simps})
also have \( \ldots \leq (\ln 3 + 1) / 3 \) using \( \text{int exp-half} \)
  by (intro decreasing-sum-le-integral \text{divide-nonneg-pos mono}) (auto simp: powr-minus field-simps)
finally have \( \text{-Re (deriv zeta 2)} \leq (16 * \ln 3 + 9 * \ln 2 + 12) / 36 \)
  by simp
also have \( \ln 3 \leq (11 / 10 : \text{real}) \)
  using \( \text{ln-approx-bounds[of \{3 \}]} \) by (simp add: power-numeral-reduce numeral-2-eq2)
  hence \((16 * \ln 3 + 9 * \ln 2 + 12) / 36 \leq (16 * (11 / 10) + 9 + 5 / 36 + 12) / (36 :: \text{real}) \)
  using \( \text{ln2-le-25-over-36} \) by (intro add mono mult-left mono divide-right mono)
auto
also have \( \ldots < 1 \) by simp

103
finally show 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** $x :: \text{real}$

**defines** $R \equiv (\sum (p, i) | \text{prime } p \land i > 1 \land \text{real } (p^i) \leq x. \ln (\text{real } p) / p^i)$

**shows** $R < \frac{9}{\pi^2}$

**proof**

- **define** $S'$ where $S' = \{(p, i). \text{prime } p \land i > 1 \land \text{real } (p^i) \leq x\}$
- **define** $S''$ where $S'' = \{(p, i). \text{prime } p \land i > 1 \land \text{real } (p^{\text{Suc } i}) \leq x\}$

**have** finite-row: finite \{i. i > 1 \land \text{real } (p^i) \leq x\} if p: prime p for p k

**proof** (rule finite-subset)

  - **show** \{i. i > 1 \land \text{real } (p^i) \leq x\} \subseteq {..nat [x]}
  - **proof** safe
    - **fix** i assume i: i > 1 real (p^i) \leq x
    - **have** i < 2^i by (induction i) auto
      - **also from** p **have** ... \leq p^i by (intro power-mono) (auto dest: prime-gt-1-nat)
      - **also have** real ... \leq x using i by simp
        - **finally show** i \leq nat [x] by linarith
  - **qed**
  - **qed auto**

**have** $S'' \subseteq S'$ unfolding $S''$-def $S'$-def

**proof** safe

  - **fix** p i assume pi: prime p real (p^Suc i) \leq x i > 1
  - **have** real (p^Suc i) \leq \text{real } (p^i)
    - **using** pi unfolding of-nat-le-iff by (intro power-increasing) (auto dest: prime-gt-1-nat)
    - **also have** ... \leq x by fact
      - **finally show** real (p^i) \leq x .
  - **qed**

**have** $S'$-alt: $S' = (SIGMA p:\{p. \text{prime } p \land \text{real } p \leq x\}. \{i. i > 1 \land \text{real } (p^i) \leq x\})$

**unfolding** $S'$-def

**proof** safe

  - **fix** p i assume prime p real (p^i) \leq x i > 1
  - **hence** p^i \leq p^i
    - **by** (intro power-increasing) (auto dest: prime-gt-1-nat)
    - **also have** real ... \leq x by fact
      - **finally show** real p \leq x by simp
qed

have finite: finite \{ p, prime p ∧ real p ≤ 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 ≤ 3 / 2 * (\sum (p, i) \in S' ∧ even i. ln (real p) / real (p ^ i))
proof -
  have R = (\sum y \in \{0, 1\}. \sum z \in S' ∧ snd z mod 2 = y. ln (real (fst z)) / real (fst z ^ snd z))
    using finite S' by (subst sum) (auto simp: case-prod-unfold R-def S'-def)
  also have \ldots = (\sum (p, i) \in S' ∧ even i. ln (real p) / real (p ^ i)) +
    (\sum (p, i) \in S' ∧ 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 \ldots = (1 / 2) * (\sum (p, i) \in S' ∧ even i. ln (real p) / real (p ^ i))
    by (intro sum even-weight) (auto simp: case-prod-unfold)
  also have \ldots ≤ (\sum (p, i) \in S' ∧ even i. ln (real p) / real (p ^ i) + \ldots =
    3 / 2 * (\sum (p, i) \in S' ∧ even i. ln (real p) / real (p ^ i))
    by simp
finally show ?thesis by simp

qed
also have \( \ldots \leq (\sum p \mid \text{prime } p \land \text{real } p \leq x. \ln (\text{real } p) / (\text{real } p^2 - 1)) \)

proof (rule sum-mono)

fix \( p \) assume \( p: p \in \{ p. \text{prime } p \land \text{real } p \leq x \} \)

have \( \sum i \mid i > 0 \land \text{even } i \land \text{real } (p^i) \leq x. (1 / \text{real } p) - i \) =

\( \sum i \mid \text{real } (p^i (2 \cdot i + 2)) \leq x. (1 / \text{real } p) - (2 \cdot i) / \text{real } p^2 \) (is \( - = ?S / - \) unfolding sum-divide-distrib)

by (rule sum.reindex-bij-witness[of \( - \) : \( \lambda i. 2 \cdot \text{Suc } i \cdot \text{Suc } (i - 2) \) \( \text{div } 2 \))

(insert \( p > 1 \), auto simp: numeral-3-eq-3 power2-eq-square power-diff

\( \text{algebra-simps elim!: evenE} \))

also have \( ?S = (\sum i \mid \text{real } (p^i (2 \cdot i + 2)) \leq x. (1 / \text{real } p) - i) \)

by (subst power-mul) (simp-all add: algebra-simps power-divide)

also have \( \ldots \leq (\sum i. (1 / \text{real } p^2) - i) \)

using exponents-le-finite[of \( 2 \cdot 2 \cdot x \) \( p > 1 \)]

by (intro sum-le-suminf) (auto simp: summable-geometric-iff)

also have \( \ldots = \text{real } p^2 / (\text{real } p^2 - 1) \)

using \( \langle p > 1 \rangle \) by (subst suminf-geometric) (auto simp: field-simps)

also have \( \ldots = \text{real } p^2 = 1 / (\text{real } p^2 - 1) \)

using \( \langle p > 1 \rangle \) by (simp add: divide-simps)

finally have \( (\sum i \mid 0 < i \land \text{even } i \land \text{real } (p^i) \leq x. (1 / \text{real } p) - i) \leq 1 / (\text{real } p^2 - 1) \) (is \( ?lhs \leq ?rhs \))

using \( \langle p > 1 \rangle \) by (simp add: divide-right-mono)

thus \( \ln (\text{real } p) \leq ?rhs \leq \ln (\text{real } p) / (\text{real } p^2 - 1) \)

using \( \langle p > 1 \rangle \) by (simp add: divide-simps)

qed

also have \( \ldots = (\sum a \mid \text{prime } a \land \text{real } a \leq x. \ln (\text{real } a) / (\text{real } a^2 - 1)) \)

using finite by (intro infsetsum-finite [symmetric]) auto

also have \( \ldots \leq (\sum a \mid \text{prime } a. \ln (\text{real } a) / (\text{real } a^2 - 1)) \)

using eval-fds-logderiv-zeta-real[of \( 2 \)] finite

by (intro infsetsum-mono-neutral-left divide-nonneg-pos) (auto simp: dest: prime-gl-1-nat)

also have \( \ldots = -\text{Re } (\text{deriv zeta} \ (\text{of-real } 2) \ / \text{zeta} \ (\text{of-real } 2)) \)

by (subst eval-fds-logderiv-zeta-real) auto

also have \( \ldots = (-\text{Re } (\text{deriv zeta } 2)) * (6 / \pi^2) \)

by (simp add: zeta-even-numeral)

also have \( \ldots < 1 * (6 / \pi^2) \)

using deriv-zeta-2-bound by (intro mult-strict-right-mono) auto

also have \( 3 / 2 \ldots = 9 / \pi^2 \) by simp

finally show \( ?thesis \) by simp

qed

We now consider the equation

\[
\ln(n!) = \sum_{k \leq n} \Lambda(k) \left[ \frac{n}{k} \right]
\]

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: \( \mathfrak{M}(n) - \ln(n) \in (-1 - \frac{9}{\pi^2}, \ln 4] \)

\[ \text{theorem mertens-bound-strong:} \]
\[ \text{fixes } n :: \text{nat} \text{ assumes } n : n > 0 \]
\[ \text{shows } \mathfrak{M}(n) - \ln(n) \in \{-1 - \frac{9}{\pi^2}, \ln 4\} \]
\[ \text{proof (cases } n \geq 3) \]
\[ \text{case } False \]
\[ \text{with } n \text{ consider } n = 1 | n = 2 \text{ by force} \]
\[ \text{thus } \exists \text{thesis} \]
\[ \text{proof cases} \]
\[ \text{assume [simp]: } n = 1 \]
\[ \text{have } -1 + (-9 / \pi^2) < 0 \]
\[ \text{by (intro add-neg-neg divide-neg-pos) auto} \]
\[ \text{thus } \exists \text{thesis by simp} \]
\[ \text{next} \]
\[ \text{assume [simp]: } n = 2 \]
\[ \text{have eq: } \mathfrak{M}(n) - \ln(n) = -\ln 2 / 2 \text{ by (simp add: eval-M)} \]
\[ \text{have } -1 - 9 / \pi^2 + \ln 2 / 2 \leq -1 - 9 / 4 + 2 + 25 / 36 / 2 \]
\[ \text{using pi-less-4 ln2-le-25-over-36} \]
\[ \text{by (intro diff-mono add-mono divide-left-mono power-mono) auto} \]
\[ \text{also have } \ldots < 0 \text{ by simp} \]
\[ \text{finally have } -\ln 2 / 2 > -1 - 9 / \pi^2 \text{ by simp} \]
\[ \text{moreover} \{ \]
\[ \text{have } -\ln 2 / 2 \leq (0::real) \text{ by (intro divide-nonpos-pos) auto} \]
\[ \text{also have } \ldots \leq \ln 4 \text{ by simp} \]
\[ \text{finally have } -\ln 2 / 2 \leq \ln (4 :: real) \text{ by simp} \]
\[ \} \]
\[ \text{ultimately show } \exists \text{thesis unfolding eq by simp} \]
\[ \text{qed} \]

\[ \text{next} \]
\[ \text{case } True \]
\[ \text{hence } n : n \geq 3 \text{ by simp} \]
\[ \text{have finite: finite \{(p, i). prime p \land i \geq 1 \land p ^ i \leq n\} \}
\[ \text{proof (rule finite-subset)} \]
\[ \text{show \{(p, i). prime p \land i \geq 1 \land p ^ i \leq n\} \]
\[ \subseteq \{(..nat [root 1 (real n)] \times \{..nat [log 2 (real n)]\}\}
\[ \text{using primepowls-subset[of real n 1] n unfolding of-nat-le-iff by auto} \]
\[ \text{qed auto} \]
\[ \text{define } r \text{ where } r = \text{prime-sum-upto (\lambda p. \ln (real p) * frac (real n / real p)) } n \]

107
\begin{verbatim}
define R where R = (∑(p,i) | prime p ∧ i > 1 ∧ p ^ i ≤ n. ln (real p) * real (n div (p ^ i))))
define R' where R' = (∑(p,i) | prime p ∧ i > 1 ∧ p ^ i ≤ n. ln (real p) / p ^ i)

have \{simp\}: ln (4 :: real) = 2 * ln 2
  using ln-realpow[of 2 2] by simp

from pi-less-4 have ln pi ≤ ln 4 by (subst ln-le-cancel-iff) auto
also have ... = 2 * ln 2 by simp
also have ... ≤ 2 * (25 / 36) by (intro mult-left-mono ln2-le-25-over-36) auto
finally have ln-pi : ln pi ≤ 25 / 18 by simp
have ln 3 ≤ ln (4::nat) by (subst ln-le-cancel-iff) auto
also have ... = 2 * ln 2 by simp
also have ... ≤ 2 * (25 / 36) by (intro mult-left-mono ln2-le-25-over-36) auto
finally have ln3 : ln (3::real) ≤ 25 / 18 by simp

have R / n = (∑(p,i) | prime p ∧ i > 1 ∧ p ^ i ≤ n. ln (real p) * (real (n div (p ^ i)) / n))
  by (simp add: R-def sum-divide-distrib field-simps case-prod-unfold)
also have ... ≤ (∑(p,i) | prime p ∧ i > 1 ∧ p ^ i ≤ n. ln (real p) * (1 / p ^ i))
unfolding R'-def case-prod-unfold using n
  by (intro sum-mono mult-left-mono) (auto simp: field-simps real-of-nat-div dest: prime-gt-0-nat)
also have ... = R' by (simp add: R'-def)
also have R' < 9 / pi^2
  unfolding R'-def using mertens-remainder-aux-bound[of n] by simp
finally have R / n < 9 / pi^2
moreover have R ≥ 0
unfolding R-def by (intro sum-nonneg mult-nonneg-nonneg) (auto dest: prime-gt-0-nat)
ultimately have R-bounds: R / n ∈ {0..<9 / pi^2} by simp

have ln (fact n :: real) ≤ ln (2 * pi * n) / 2 + n * ln n - n + 1 / (12 * n)
  using ln-fact-bounds[of n] n by simp
also have ... / n = -1 + (ln 2 + ln pi) / (2 * n) + (ln n / n) / 2 + 1 / (12 * real n ^ 2)
  using n by (simp add: power2-eq-square field-simps ln-mult)
also have ... ≤ -1 + (ln 2 + ln pi) / (2 * 3) + (ln 3 / 3) / 2 + 1 / (12 * 3^2)
  using exp-le n pi-gt3
by (intro add-mono divide-right-mono divide-left-mono mult-mono
  mult-pos-pos ln-x-over-x-mono power-mono) auto
also have ... ≤ -1 + (25 / 36 + 25 / 18) / (2 * 3) + (25 / 18 / 3) / 2 + 1 / (12 * 3^2)
  using ln-pi ln2-le-25-over-36 ln3 by (intro add-mono divide-left-mono divide-right-mono) auto
also have ... ≤ 0 by simp
finally have ln n = ln (fact n) / n ≥ 0 using n by (simp add: divide-right-mono)
have -ln (fact n) ≤ -ln (2 * pi * n) / 2 - n * ln n + n
  using ln-fact-bounds(1)[of n] n by simp
\end{verbatim}
also have \( \ln n + \ldots / n = -\ln (2 * pi) / (2 * n) - (\ln n / n) / 2 + 1 \)
using \( n \) by (simp add: field-simps ln-mult)
also have \( \ldots \leq 0 - 0 + 1 \)
using \( pi-gl3 n \) by (intro add-mono diff-mono) auto
finally have upper: \( \ln n - \ln (\text{fact } n) / n \leq 1 \)
using \( n \) by (simp add: divide-right-mono)
with \( \ln n - \ln (\text{fact } n) / n \geq 0 \) have fact-bounds: \( \ln n - \ln (\text{fact } n) / n \in \{0..1\} \) by simp

have \( r \leq \text{prime-sum-upto} (\lambda p. \ln p * 1) n \)
using less-imp-le[OF frac-lt-1] unfolding r-def \( \theta \)-def prime-sum-upto-def
by (intro sum-mono mult-left-mono) (auto simp: dest: prime-gl-0-nat)
also have \( \ldots = \theta n \) by (simp add: \( \theta \)-def)
also have \( \ldots < \ln 4 * n \) using \( n \) by (intro \( \theta \)-upper-bound) auto
finally have \( r / n < \ln 4 \) using \( n \) by (simp add: field-simps)
moreover have \( r \geq 0 \) unfolding r-def prime-sum-upto-def
by (intro sum-nonneg mult-nonneg-nonneg) (auto dest: prime-gl-0-nat)
ultimately have r-bounds: \( r / n \in \{0..<\ln 4\} \) by simp

have \( \ln (\text{fact } n :: \text{real}) = \text{sum-upto} (\lambda k. \text{mangoldt } k * \text{real } (n \text{ div } k)) \) (real n)
by (simp add: ln-fact-conv-sum-upto-mangoldt)
also have \( \ldots = (\sum (p,i)) \) prime \( p \wedge i > 0 \wedge \text{real } (p ^ i) \leq \text{real } n. \ln (\text{real } p) * \text{real } (n \text{ div } (p ^ i)) \)
by (intro sum-upto-primepow) (auto simp: mangoldt-non-primepow)
also have \( \{(p, i). \text{prime } p \wedge i > 0 \wedge \text{real } (p ^ i) \leq \text{real } n\} = \)
\( \{(p, i). \text{prime } p \wedge i = 1 \wedge p \leq n\} \cup \)
\( \{(p, i). \text{prime } p \wedge i > 1 \wedge (p ^ i) \leq n\} \) unfolding of-nat-le-iff
by (auto simp: not-less le-Suc-eq)
also have \( (\sum (p,i)\ldots = \ln (\text{real } p) * \text{real } (n \text{ div } (p ^ i))) = \)
\( (\sum (p,i) \mid \text{prime } p \wedge i = 1 \wedge p \leq n. \ln (\text{real } p) * \text{real } (n \text{ div } (p ^ i))) \)
+ \( R \)
(is - = ?S + -)
by (subst sum.union_disjoint) (auto intro!: finite_subset[OF - finite] simp: R-def)
also have \( ?S = \text{prime-sum-upto} (\lambda p. \ln (\text{real } p) * \text{real } (n \text{ div } p)) \) n
unfolding prime-sum-upto-def
by (intro sum.reindex-bij-witness[of - \lambda p. (p, 1) fst]) auto
also have \( \ldots = \text{prime-sum-upto} (\lambda p. \ln (\text{real } p) * \text{real } n / \text{real } p) \) n - r
unfolding r-def prime-sum-upto-def sum-subtract[OF symmetric] using n
by (intro sum.cong) (auto simp: frac-def real-of-nat-div algebra_simps)
also have \( \text{prime-sum-upto} (\lambda p. \ln (\text{real } p) * \text{real } n / \text{real } p) \) n = n * \( \mathbb{M} \) n
by (simp add: primes-M-def sum-distrib-left sum-distrib-right prime-sum-upto-def field-simps)
finally have \( \mathbb{M} n - \ln n \leq \ln (\text{fact } n) / n - \ln n + r / n - R / n \)
using \( n \) by (simp add: field-simps)
hence \( \ln n - \mathbb{M} n = \ln n - \ln (\text{fact } n) / n - r / n + R / n \)
by simp
with fact-bounds r-bounds R-bounds show \( \mathbb{M} n - \ln n \in \{-1 - 9 / p \leq < \ln 4\} \)
by simp
As a simple corollary, we obtain a similar bound on the reals.

**lemma** mertens-bound-real-strong:
  
  ```
  fixes x :: real
  assumes x: x ≥ 1
  shows M x - ln x ∈ {-1 - 9 / pi ^ 2 - ln (1 + frac x / real (nat ⌊x⌋))} <.. ln 4}
  ```

**proof**
  
  ```
  have M x - ln x ≤ M (real (nat ⌊x⌋)) - ln (real (nat ⌊x⌋))
  using assms by simp
  also have ... ≤ ln 4
  using mertens-bound-strong[of nat ⌊x⌋] assms by simp
  finally have M x - ln x ≤ ln 4 .
  ```

from assms have pos: real-of-int ⌊x⌋ ≠ 0 by linarith
have frac x / real (nat ⌊x⌋) ≥ 0
  using assms by (intro divide-nonneg-pos) auto
moreover have frac x / real (nat ⌊x⌋) ≤ 1 / 1
  using assms frac-at-1[of x] by (intro frac-le) auto
ultimately have *: frac x / real (nat ⌊x⌋) ∈ {0..1} by auto
have ln x - ln (real (nat ⌊x⌋)) = ln (x / real (nat ⌊x⌋))
  using assms by (subst ln-div) auto
also have x / real (nat ⌊x⌋) = 1 + frac x / real (nat ⌊x⌋)
  using assms pos by (simp add: frac-def field-simps)
finally have M x - ln x > -1-9/pi^2-ln (1 + frac x / real (nat ⌊x⌋))
  using mertens-bound-strong[of nat ⌊x⌋] x by simp
with M x - ln x ≤ ln 4; show ?thesis by simp

We weaken this estimate a bit to obtain nicer bounds:

**lemma** mertens-bound-real'::
  
  ```
  fixes x :: real
  assumes x: x ≥ 1
  shows M x - ln x ∈ {-2<..25/18}
  ```

**proof**
  
  ```
  have M x - ln x ≤ ln 4
  using mertens-bound-real-strong[of x] x by simp
  also have ... ≤ 25 / 18
  using ln-realpow[of 2 2] ln2-le-25-over-36 by simp
  finally have M x - ln x ≤ 25 / 18 .
  ```

have ln2: ln (2 :: real) ∈ {2/3..25/36}
  using ln-approx-bounds[of 2 1] by (simp add: eval-nat-numeral)
have ln3: ln (3::real) ∈ {1..10/9}
  using ln-approx-bounds[of 3 1] by (simp add: eval-nat-numeral)
have ln5: ln (5::real) ∈ {4/3..76/45}
  using ln-approx-bounds[of 5 1] by (simp add: eval-nat-numeral)
have ln7: ln (7::real) ∈ {3/2..15/7}
  using ln-approx-bounds[of 7 1] by (simp add: eval-nat-numeral)
have ln11: ln (11::real) ∈ {5/3..290/99}
— 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.

\[
\begin{align*}
\text{have } & M \cdot x - \ln x > -2 \\
\text{proof } & \ (\text{cases } x \geq 11) \\
\text{case } & \text{False} \\
\text{hence } & x \in \{1..<2\} \lor x \in \{2..<3\} \lor x \in \{3..<5\} \lor x \in \{5..<7\} \lor x \in \{7..<11\} \\
\text{using } & x \text{ by force} \\
\text{thus } & \text{?thesis} \\
\text{proof } & (\text{elim disjE})
\end{align*}
\]

\[
\begin{align*}
& \text{assume } x: x \in \{1..<2\} \\
\text{hence } & \ln x - M \cdot x \leq \ln 2 - 0 \\
& \text{by } (\text{intro diff-mono}) \text{ auto} \\
& \text{also have } \ldots < 2 \text{ using ln2-le-25-over-36 by simp} \\
\text{finally show } & \text{?thesis} \text{ by simp}
\end{align*}
\]

next
\[
\begin{align*}
& \text{assume } x: x \in \{2..<3\} \\
\text{hence } & [\text{simp}]: x = 2 \text{ by } (\text{intro floor-unique}) \text{ auto} \\
& \text{from } x \text{ have } \ln x - M \cdot x \leq \ln 3 - \ln 2 / 2 \\
& \text{by } (\text{intro diff-mono}) \text{ (auto simp: eval-M)} \\
& \text{also have } \ldots = \ln (9 / 2) / 2 \text{ using ln-realpow[of 3 2] by simp} \\
& \text{also have } \ldots < 2 \text{ using ln-approx-bounds[of 9 / 2 1] by simp add: eval-nat-numeral} \\
\text{finally show } & \text{?thesis} \text{ by simp}
\end{align*}
\]

next
\[
\begin{align*}
& \text{assume } x: x \in \{3..<5\} \\
\text{hence } & M \cdot 3 = M \cdot x \\
\text{unfolding } & \text{primes-M-def} \\
& \text{by } (\text{intro prime-sum-upto-eqI[where } a' = 3 \text{ and } b' = 4]) \\
& \text{ (auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)} \\
& \text{also have } M \cdot 3 = \ln 2 + 2 + \ln 3 / 3 \\
& \text{by } (\text{simp add: eval-M eval-nat-numeral mark-out-code}) \\
\text{finally have } & [\text{simp}]: M \cdot x = \ln 2 / 2 + \ln 3 / 3 .. \\
& \text{from } x \text{ have } \ln x - M \cdot x \leq \ln 5 - (\ln 2 / 2 + \ln 3 / 3) \\
& \text{by } (\text{intro diff-mono}) \text{ auto} \\
& \text{also have } \ldots < 2 \text{ using ln2 ln3 ln5 by simp} \\
\text{finally show } & \text{?thesis} \text{ by simp}
\end{align*}
\]

next
\[
\begin{align*}
& \text{assume } x: x \in \{5..<7\} \\
\text{hence } & M \cdot 5 = M \cdot x \\
\text{unfolding } & \text{primes-M-def} \\
& \text{by } (\text{intro prime-sum-upto-eqI[where } a' = 5 \text{ and } b' = 6]) \\
& \text{ (auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)} \\
& \text{also have } M \cdot 5 = \ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5 \\
& \text{by } (\text{simp add: eval-M eval-nat-numeral mark-out-code})
\end{align*}
\]
finally have \([\text{simp}]\): \(\mathcal{M} x = \ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5 \ldots\)

from \(x\) have \(\ln x - \mathcal{M} x \leq \ln 7 - (\ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5)\)
  by (intro diff-mono) auto
also have \(\ldots < 2\) using ln2 ln3 ln5 ln7 by simp
finally show \(?thesis\) by simp

next
assume \(x: x \in \{7..<11\}\)
hence \(\mathcal{M} 7 = \mathcal{M} x\)
  unfolding primes-M-def
  by (intro prime-sum-upto-eqI \[where \(a' = 7\) and \(b' = 10\)])
  (auto simp: eval-M eval-nat-numeral mark-out-code)
also have \(\mathcal{M} 7 = \ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5 + \ln 7 / 7 \ldots\)
  by (simp add: eval-nat-numeral mark-out-code)
finally have \([\text{simp}]\): \(\mathcal{M} x = \ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5 + \ln 7 / 7 \ldots\)
from \(x\) have \(\ln x - \mathcal{M} x \leq \ln 11 - (\ln 2 / 2 + \ln 3 / 3 + \ln 5 / 5 + \ln 7 / 7 \ldots)\)
  by (intro diff-mono) auto
also have \(\ldots < 2\) using ln2 ln3 ln5 ln7 ln11 by simp
finally show \(?thesis\) by simp
qed

next
case \(\text{True}\)
have \(\ln x - \mathcal{M} x \leq 1 + 9/\pi^2 + \ln (1 + \text{frac} x / \text{real (nat [x])})\)
  using mertens-bound-real-strong[of \(x\)] \(x\) by simp
also have \(1 + \text{frac} x / \text{real (nat [x])} \leq 1 + 1 / 11\)
  using \(\text{True frac-le-1[of x]}\) by (intro add-mono frac-le) auto
hence \(\ln (1 + \text{frac} x / \text{real (nat [x])}) \leq \ln (1 + 1 / 11)\)
  using \(x\) by (subst ln-le-cancel-iff) (auto intro!: add-pos-nonneg)
also have \(\ldots = \ln (12 / 11)\) by simp
also have \(\ldots \leq 1585 / 18216\)
  using ln-approx-bounds[of \(12 / 11\) \(1\)] by (simp add: eval-nat-numeral)
also have \(9 / \pi^2 \leq 9 / 3.141592653588 \leq 2\)
  using pi-approx by (intro divide-left-mono power-mono mult-pos-pos) auto
also have \(1 + \ldots + 1585 / 18216 < 2\)
  by (simp add: power2-eq-square)
finally show \(?thesis\) by simp
qed

with \(\mathcal{M} x - \ln x \leq 25 / 18\) show \(?thesis\) by simp
qed

corollary mertens-first-theorem:
  fixes \(x::\text{real}\) assumes \(x: x \geq 1\)
  shows \(|\mathcal{M} x - \ln x| < 2\)
  using mertens-bound-real[of \(x\)] \(x\) by (simp add: abs-if)
5.2 Mertens’ Second Theorem

Mertens’ Second Theorem concerns the asymptotics of the Prime Harmonic Series, namely

$$\sum_{p \leq x} \frac{1}{p} = \ln \ln x + M + O\left(\frac{1}{\ln x}\right)$$

where $M \approx 0.261497$ 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. (\Re 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: (a::real) > 1
  shows set-integrable borel {a<..} (\lambda t. 1 / (t * ln t ^ 2)) (is ?th1)
    and set-lebesgue-integral borel {a<..} (\lambda t. 1 / (t * ln t ^ 2)) = 1 / ln a (is ?th2)
    and ((\lambda t. 1 / (t * (ln t)^2)) has-integral 1 / ln a) {a<..} (is ?th3)

proof –
  have cont: isCont (\lambda t. 1 / (t * (ln t)^2)) x if x > a for x
    using that a by (auto intro: continuous-intros)
  have deriv: ((\lambda x. -1 / ln x) has-real-derivative 1 / (x * (ln x)^2)) (at x) if x > a for x
    using that a by (auto intro: derivative-eq-intros simp: power2-eq-square field-simps)
  have lim1: ((\lambda x. -1 / ln x) o real-of-ereal) ----> (1 / ln a) (at-right (ereal a))
    unfolding ereal-tendsto-simps using a by (real-asympt simp: field-simps)
  have lim2: ((\lambda x. -1 / ln x) o real-of-ereal) ----> 0) (at-left (ereal a))
    unfolding ereal-tendsto-simps using a by (real-asympt simp: field-simps)
  have set-integrable borel (cinterval a \infty) (\lambda t. 1 / (t * (ln t)^2))
    by (rule interval-integral-FTC-nonneg[OF - deriv cont - lim1 lim2]) (use a in auto)
  thus ?th1 by simp
  have interval-lebesgue-integral borel (ereal a) \infty (\lambda t. 1 / (t * (ln t)^2)) = 0 -
    (-(1 / ln a))
    by (rule interval-integral-FTC-nonneg[OF - deriv cont - lim1 lim2]) (use a in auto)
  thus ?th2 by (simp add: interval-integral-to-infinity-eq)

have ((\lambda t. 1 / (t * (ln t)^2)) has-integral
  set-lebesgue-integral lebesgue {a<..} (\lambda t. 1 / (t * (ln t)^2)) {a<..}
  using ?th1) by (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)^2)) = 1 / ln a

113
using ⟨?th2⟩ unfolding set-lebesgue-integral-def by (subst integral-completion)
auto

finally show ?th3.
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 :: real)
defines r ≡ (λt. (∃ t − ln t) / (t * ln t ^ 2))
shows integrable-meissel-mertens: set-integrable lborel {a<..} r
  and meissel-mertens-integral-le: norm (integral {a<..} r) ≤ 2 / ln a
proof
  have *: ((λt. 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)
  then show set-integrable lborel {a<..} r unfolding set-integrable-def
    proof (rule Bochner-Integration.integrable-bound[OF - - AE-I2])
      have integrable lborel (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 integrable-on {a<..} r
        by (simp add: algebra-simps)
      thus integrable lborel (λt::real. indicator {a<..} t *R (2 / (t * ln t ^ 2)))
        by simp
      fix x :: real
      show norm (indicat-real {a<..} x *R r x) ≤
        norm (indicat-real {a<..} x *R (2 / (x * ln x ^ 2)))
      proof (cases x > a)
        case True
        thus ?thesis
        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)))
        (rule integral-norm-bound-integral)
      show (λx. 2 * (1 / (x * (ln x)^2))) integrable-on {a<..}
        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

114
using * by (simp add: has-integral-iff)
finally show norm (integral {a<..} r) ≤ 2 / ln a .
q ed

lemma integrable-on-meissel-mertens:
assumes A ⊆ {1..} Inf A > 1 A ∈ sets borel
shows (λt. (M 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 - 1]) auto
hence A ⊆ {Inf A..} by (auto simp: cInf_lower)
also have ... ⊆ {x<..} using x by auto
finally have *: A ⊆ {x<..} .
have set-integrable l borel A (λt. (M t − ln t) / (t * ln t ^ 2))
  by (rule set-integrable-subset[of integrable-meissel-mertens[of x]]) (use x *
assms in auto)
thus *thesis by (simp add: set-borel-integral-eq-integral(1))
q ed

lemma meissel-mertens-bounds: |meissel-mertens − 1 + ln (ln 2)| ≤ 2 / ln 2
proof –
have *: {2..} − {2<..} = {2::real} by auto
also have negligible ... by simp
finally have integral {2..} (λt. (M t − ln t) / (t * (ln t) ^ 2)) =
  integral {2<..} (λt. (M t − ln t) / (t * (ln t) ^ 2))
  by (intro sym[of integrable-meissel-mertens-negligible]) auto
also have norm ... ≤ 2 / ln 2
  by (rule meissel-mertens-integral-le) auto
finally show |meissel-mertens − 1 + ln (ln 2)| ≤ 2 / ln 2
  by (simp add: meissel-mertens-def)
q ed

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 ≡ prime-sum-upto (λp. 1 / p)
shows ∀x. x ≥ 2 ⇒ |f x − ln (ln x) − meissel-mertens| ≤ 4 / ln x
  and (λx. f x − ln (ln x) − meissel-mertens) ∈ O(λx. 1 / ln x)
proof –
define r where r = (λt. (M t − ln t) / (t * ln t ^ 2))

{ 
fix x :: real assume x: x > 2

have ((λt. M t * (−1 / (t * ln t ^ 2)))) has-integral M x * (1 / ln x) − M 2
  * (1 / ln 2) −
  (∑ n:real − {2<..} · ind prime n * (ln n / real n) * (1 / ln n)) {2..x}
unfolding primes-M-def prime-sum-upto-altdef1 using x
by (intro partial-summation-strong[of {}])

115
(auto intro!: continuous-intros derivative-eq-intros simp: power2-eq-square
  simp flip: has-real-derivative-iff-has-vector-derivative)
also have \(\mathbb{R} x \cdot (1 / \ln x) - \mathbb{R} 2 \cdot (1 / \ln 2) =
\left(\sum n \in \mathbb{R} \cdot \{2 < x\} \cdot \text{ind prime } n \cdot (\ln n / n) \right) =\)
\(\mathbb{R} x / \ln x - \left(\sum n \in \text{insert 2 (real \cdot \{2 < x\})} \cdot \text{ind prime } n \cdot (\ln n / n) \right) (1 / \ln n)\)
(is - - !?S)
by (subst sum.insert)
(auto simp: primes-M-def finite-vimage-real-of-nat-greaterThanAtMost eval-prime-sum-upto)
also have ?S = f x
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 \((\lambda t. -\mathbb{R} t / (t \cdot \ln t ^ 2)) \text{ has-integral} \ (\mathbb{R} x / \ln x - f x)\)
\{2..x\}
by simp
from has-integral-neg[OF this]
have \((\lambda t. \mathbb{R} t / (t \cdot \ln t ^ 2) - 1 / (t \cdot \ln t)) \text{ has-integral} \ (f x - \mathbb{R} x / \ln x) \{2..x\}\) by simp
hence \((\lambda t. \mathbb{R} t / (t \cdot \ln t ^ 2) - 1 / (t \cdot \ln t)) \text{ has-integral} \ (f x - \mathbb{R} x / \ln x - \ln (\ln x) - \ln (\ln 2))) \{2..x\}\) using x
by (intro has-integral-diff fundamental-theorem-of-calculus)
(auto simp flip: has-real-derivative-iff-has-vector-derivative
intro: derivative-eq-intros)
also have ?this \leftrightarrow (r \text{ has-integral} \ (f x - \mathbb{R} x / \ln x - (\ln (\ln x) - \ln (\ln 2))) \{2..x\})
by (intro has-integral-cong) (auto simp: r-def field-simps power2-eq-square)
finally have . . . .
} note integral = this

define R\(_{\mathbb{R}}\) where \(R_{\mathbb{R}} = (\lambda x. \mathbb{R} x - \ln x)\)
have \(\mathbb{R}\): \(\mathbb{R} x = \ln x + R_{\mathbb{R}} x\) for \(x\) by (simp add: R\(_{\mathbb{R}}\)-def)
define I where \(I = (\lambda x. \ln x + r)\)
define C where \(C = (1 - \ln (\ln 2) + \ln 2)\)
have C-altdef: \(C = \text{meissel-mertens}\)
by (simp add: I-def r-def C-def meissel-mertens-def)

show bound: \(\left| f x - \ln (\ln x) - \text{meissel-mertens} \right| \leq 4 / \ln x \) if \(x\): \(x \geq 2\) for \(x\)
proof (cases \(x = 2\))
case True
  hence \(\left| f x - \ln (\ln x) - \text{meissel-mertens} \right| = |1 / 2 - \ln (\ln 2) - \text{meissel-mertens}|\)
  by (simp add: f-def eval-prime-sum-upto)
also have . . \(\leq 2 / \ln 2 + 1 / 2\)
  using meissel-mertens-bounds by linarith
also have . . \(\leq 2 / \ln 2 + 2 / \ln 2\) using ln2-le-25-over-36
  by (intro add-mono divide-left-mono) auto
finally show ?thesis using True by simp

116
next
  case False
  hence \( x > 2 \) using \( x \) by simp
  have integral \( \{ 2..x \} \) \( r + I x = \text{integral} \left( \{ 2..x \} \cup \{ x.. \} \right) \) \( r \) unfolding I-def r-def using \( x \)
  by (intro integral-Un [symmetric] integrable-on-meißel-mertens) (auto simp: max-def r-def)
  also have \( \{ 2..x \} \cup \{ x.. \} = \{ 2.. \} \) using \( x \) by auto
  finally have \( \varepsilon : \text{integral} \{ 2..x \} \) \( r = I 2 - I x \) unfolding I-def by simp
  have eq: \( f x - \ln \left( \ln x \right) - C = M x / \ln x - I x \)
  using integral[of \( x \)] by (auto simp: C-def field-simps \( M \) has-integral-iff *)
  also have \( \ldots \leq \left| M x / \ln x \right| + \text{norm} \left( I x \right) \)
  unfolding real-norm-def by (rule abs-triangle-ineq)
  also have \( \left| M x / \ln x \right| \leq 2 / \left| \ln x \right| \)
  unfolding M-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 \( \text{norm} \ldots \leq 2 / \ln x \)
  using meissel-mertens-integral-le[of \( x \)] \( x \) by (simp add: r-def)
  finally show \( \lfloor f x - \ln \left( \ln x \right) - \text{meissel-mertens} \rfloor \leq 4 / \ln x \)
  using \( x \) by (simp add: C-altdef)
qed

have \( (\lambda x. f x - \ln \left( \ln x \right) - C) \in O(\lambda x. 1 / \ln x) \)
proof (intro landau-o.big[of \( 4 \)] eventually-mono[of \( \text{Eventually} \geq \text{at-top}[\text{of} 2] \)])
fix \( x :: \text{real} \) assume \( x \geq 2 \)
with \( \text{bound} \left( \text{of} x \right) \) show \( \left| f x - \ln \left( \ln x \right) - C \right| \leq 4 * \text{norm} \left( 1 / \ln x \right) \)
by (simp add: C-altdef)
qed (auto intro!: add-pos-nonneg)
thus \( (\lambda x. f x - \ln \left( \ln x \right) - \text{meissel-mertens}) \in O(\lambda x. 1 / \ln x) \)
by (simp add: C-altdef)
qed

corollary prime-harmonic-asympt-eqv: \( \text{prime-sum-upto} \left( \lambda p. 1 / p \right) \sim [\text{at-top}] \left( \lambda x. \ln \left( \ln x \right) \right) \)
proof –
  define \( f \) where \( f = \text{prime-sum-upto} \left( \lambda p. 1 / p \right) \)
  have \( (\lambda x. f x - \ln \left( \ln x \right) - \text{meissel-mertens} + \text{meissel-mertens}) \in o(\lambda x. \ln \left( \ln x \right)) \)
  unfolding f-def
  by (rule sum-in-smallo[of \( \text{Landau-o.big-small-trans}[\text{of} \text{mertens-second-theorem}(2)] \)])
  real-asympt+
  hence \( (\lambda x. f x - \ln \left( \ln x \right)) \in o(\lambda x. \ln \left( \ln x \right)) \)
  by simp
  thus \( \text{thesis} \) unfolding f-def
  by (rule smallo-imp-asympt-eqv)
qed
As a corollary, we get the divergence of the prime harmonic series.

\textbf{corollary prime-harmonic-diverges: filterlim (prime-sum-upto (λp . 1 / p)) at-top at-top using asymp-equiv-sym[OF prime-harmonic-asymp-equiv] by (rule asymp-equiv-at-top-transfer) real-asymp}

end

6 Acknowledgements

Paulson was supported by the ERC Advanced Grant ALEXANDRIA (Project 742178) funded by the European Research Council at the University of Cambridge, UK.

References


