

The Prime Number Theorem

Manuel Eberl and Larry Paulson

December 14, 2021

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 ϑ and ψ 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	47
3	Prime-Counting Functions	62
3.1	Definitions	63
3.2	Basic properties	66
3.3	The n -th prime number	67
3.4	Relations between different prime-counting functions	72
3.5	Bounds	77
3.6	Equivalence of various forms of the Prime Number Theorem	84
3.7	The asymptotic form of Mertens' First Theorem	89
4	The Prime Number Theorem	94
4.1	Constructing Newman's function	95
4.2	The asymptotic expansion of \mathfrak{M}	103
4.3	The asymptotics of the prime-counting functions	109
5	Mertens' Theorems	112
5.1	Absolute Bounds for Mertens' First Theorem	112
5.2	Mertens' Second Theorem	123
6	Acknowledgements	128

1 Auxiliary material

theory *Prime-Number-Theorem-Library*

imports

Zeta-Function.Zeta-Function

HOL-Real-Asymp.Real-Asymp

begin

lemma *asymp-equivD-strong*:

assumes $f \sim [F] g$ *eventually* $(\lambda x. f\ x \neq 0 \vee g\ x \neq 0)$ F

shows $((\lambda x. f\ x / g\ x) \longrightarrow 1)$ F

proof –

from *assms*(1) **have** $((\lambda x. \text{if } f\ x = 0 \wedge g\ x = 0 \text{ then } 1 \text{ else } f\ x / g\ x) \longrightarrow 1)$ F

by (*rule asymp-equivD*)

also have $?this \longleftrightarrow ?thesis$

by (*intro filterlim-cong eventually-mono[OF assms(2)] auto*)

finally show $?thesis$.

qed

lemma *frontier-real-Ici* [*simp*]:

fixes $a :: \text{real}$

shows $\text{frontier } \{a..\} = \{a\}$

unfolding *frontier-def* **by** (*auto simp: interior-real-atLeast*)

lemma *sum-upto-ln-conv-sum-upto-mangoldt*:

sum-upto $(\lambda n. \ln (\text{real } n))\ x = \text{sum-upto } (\lambda n. \text{mangoldt } n * \text{nat } \lfloor x / \text{real } n \rfloor)\ x$

proof –

have *sum-upto* $(\lambda n. \ln (\text{real } n))\ x =$

sum-upto $(\lambda n. \sum d \mid d \text{ dvd } n. \text{mangoldt } d)\ x$

by (*intro sum-upto-cong*) (*simp-all add: mangoldt-sum*)

also have $\dots = \text{sum-upto } (\lambda k. \text{sum-upto } (\lambda d. \text{mangoldt } d)\ (x / \text{real } k))\ x$

by (*rule sum-upto-sum-divisors*)

also have $\dots = \text{sum-upto } (\lambda n. \text{mangoldt } n * \text{nat } \lfloor x / \text{real } n \rfloor)\ x$

unfolding *sum-upto-altdef* **by** (*simp add: mult-ac*)

finally show $?thesis$.

qed

lemma *ln-fact-conv-sum-upto-mangoldt*:

$\ln (\text{fact } n) = \text{sum-upto } (\lambda k. \text{mangoldt } k * (n \text{ div } k))\ n$

proof –

have [*simp*]: $\{0 <.. \text{Suc } n\} = \text{insert } (\text{Suc } n)\ \{0 <.. n\}$ **for** n **by** *auto*

have $\ln (\text{fact } n) = \text{sum-upto } (\lambda n. \ln (\text{real } n))\ n$

by (*induction n*) (*auto simp: sum-upto-altdef nat-add-distrib ln-mult*)

also have $\dots = \text{sum-upto } (\lambda k. \text{mangoldt } k * (n \text{ div } k))\ n$

unfolding *sum-upto-ln-conv-sum-upto-mangoldt*

by (*intro sum-upto-cong*) (*auto simp: floor-divide-of-nat-eq*)

finally show $?thesis$.

qed

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

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

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

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

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

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

lemma *fds-converges-mult-const-left* [intro]:
 $fds\text{-converges } f \ s \implies fds\text{-converges } (fds\text{-const } c * f) \ s$
by (auto simp: *fds-converges-def* dest: *summable-mult*[of - c])

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

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

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

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

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

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

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

lemma *bounded-coeffs-imp-fds-abs-converges*:
fixes $s :: 'a :: dirichlet\text{-series}$ **and** $f :: 'a \ fds$
assumes $Bseq (fds\text{-nth } f) \ s \cdot 1 > 1$
shows $fds\text{-abs-converges } f \ s$
proof –
from *assms* **obtain** C **where** $C: \bigwedge n. norm (fds\text{-nth } f \ n) \leq C$
by (auto simp: *Bseq-def*)
show ?thesis
proof (rule *fds-abs-converges-comparison-test*)
from $\langle s \cdot 1 > 1 \rangle$ **show** $fds\text{-converges } (C *_{\mathbb{R}} fds\text{-zeta}) \ (s \cdot 1)$
by (intro *fds-abs-converges-imp-converges*) auto
from C **show** *eventually* $(\lambda n. norm (fds\text{-nth } f \ n) \leq fds\text{-nth } (C *_{\mathbb{R}} fds\text{-zeta}) \ n)$
at-top

by (intro always-eventually) (auto simp: fds-nth-zeta)
qed
qed

lemma bounded-coeffs-imp-fds-abs-converges':
fixes $s :: 'a :: \text{dirichlet-series}$ **and** $f :: 'a \text{ fds}$
assumes $Bseq (\lambda n. \text{fds-nth } f \ n * \text{nat-power } n \ s0) \ s \cdot 1 > 1 - s0 \cdot 1$
shows $\text{fds-abs-converges } f \ s$
proof –
have $\text{fds-nth } (\text{fds-shift } s0 \ f) = (\lambda n. \text{fds-nth } f \ n * \text{nat-power } n \ s0)$
by (auto simp: fun-eq-iff)
with **assms** **have** $Bseq (\text{fds-nth } (\text{fds-shift } s0 \ f))$ **by** *simp*
with **assms**(2) **have** $\text{fds-abs-converges } (\text{fds-shift } s0 \ f) \ (s + s0)$
by (intro bounded-coeffs-imp-fds-abs-converges) (auto simp: algebra-simps)
thus ?thesis **by** *simp*
qed

lemma bounded-coeffs-imp-abs-conv-abscissa-le:
fixes $s :: 'a :: \text{dirichlet-series}$ **and** $f :: 'a \text{ fds}$ **and** $c :: \text{ereal}$
assumes $Bseq (\lambda n. \text{fds-nth } f \ n * \text{nat-power } n \ s) \ 1 - s \cdot 1 \leq c$
shows $\text{abs-conv-abscissa } f \leq c$
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** $\dots < \text{ereal } x$ **by** *fact*
finally **have** $1 - s \cdot 1 < \text{ereal } x$ **by** *simp*
thus $\text{fds-abs-converges } f$ (of-real x)
by (intro bounded-coeffs-imp-fds-abs-converges'[OF *assms*(1)]) *auto*
qed

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

lemma
fixes $a \ b \ c :: \text{real}$
assumes $ab: a + b > 0$ **and** $c: c < -1$
shows $\text{set-integrable-powr-at-top: } (\lambda x. (b + x) \ \text{powr } c)$ *absolutely-integrable-on*
 $\{a < ..\}$

and *set-lebesgue-integral-powr-at-top*:
 $(\int x \in \{a < ..\}. ((b + x) \text{ powr } c) \partial \text{lborel}) = -((b + a) \text{ powr } (c + 1) / (c + 1))$
and *powr-has-integral-at-top*:
 $((\lambda x. (b + x) \text{ powr } c) \text{ has-integral } -((b + a) \text{ powr } (c + 1) / (c + 1)))$
 $\{a < ..\}$
proof –
let $?f = \lambda x. (b + x) \text{ powr } c$ **and** $?F = \lambda x. (b + x) \text{ powr } (c + 1) / (c + 1)$
have *limits*: $((?F \circ \text{real-of-ereal}) \longrightarrow ?F a) \text{ (at-right (ereal } a))$
 $((?F \circ \text{real-of-ereal}) \longrightarrow 0) \text{ (at-left } \infty)$
using c **ab unfolding** *ereal-tendsto-simps1* **by** $(\text{real-asymp simp: field-simps})+$
have 1 : *set-integrable lborel (einterval a ∞) ?f* **using** ab c *limits*
by $(\text{intro interval-integral-FTC-nonneg}) \text{ (auto intro!: derivative-eq-intros)}$
thus 2 : *?f absolutely-integrable-on {a < ..}*
by $(\text{auto simp: set-integrable-def integrable-completion})$
have $LBINT$ $x = \text{ereal } a .. \infty. (b + x) \text{ powr } c = 0 - ?F a$ **using** ab c *limits*
by $(\text{intro interval-integral-FTC-nonneg}) \text{ (auto intro!: derivative-eq-intros)}$
thus 3 : $(\int x \in \{a < ..\}. ((b + x) \text{ powr } c) \partial \text{lborel}) = -((b + a) \text{ powr } (c + 1) / (c + 1))$
by $(\text{simp add: interval-integral-to-infinity-eq})$
show $(?f \text{ has-integral } -((b + a) \text{ powr } (c + 1) / (c + 1))) \{a < ..\}$
using *set-borel-integral-eq-integral[OF 1] 3* **by** $(\text{simp add: has-integral-iff})$
qed

lemma *fds-converges-altdef2*:
 $fds\text{-converges } f s \iff \text{convergent } (\lambda N. \text{eval-fds } (fds\text{-truncate } N f) s)$
unfolding *fds-converges-def summable-iff-convergent'* *eval-fds-truncate*
by $(\text{auto simp: not-le intro!: convergent-cong always-eventually sum.mono-neutral-right})$

lemma *tendsto-eval-fds-truncate*:
assumes *fds-converges f s*
shows $(\lambda N. \text{eval-fds } (fds\text{-truncate } N f) s) \longrightarrow \text{eval-fds } f s$
proof –
have $(\lambda N. \text{eval-fds } (fds\text{-truncate } N f) s) \longrightarrow \text{eval-fds } f s \iff$
 $(\lambda N. \sum i \leq N. fds\text{-nth } f i / \text{nat-power } i s) \longrightarrow \text{eval-fds } f s$
unfolding *eval-fds-truncate*
by $(\text{intro filterlim-cong always-eventually allI sum.mono-neutral-left}) \text{ (auto simp: not-le)}$
also have ... **using** *assms*
by $(\text{simp add: fds-converges-iff sums-def' atLeast0AtMost})$
finally show *?thesis* .
qed

lemma *linepath-translate-left*: $\text{linepath } (c + a) (c + a) = (\lambda x. c + a) \circ \text{linepath } a b$
by $(\text{auto simp: fun-eq-iff linepath-def algebra-simps})$

lemma *linepath-translate-right*: $\text{linepath } (a + c) (b + c) = (\lambda x. x + c) \circ \text{linepath } a b$

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

lemma *integrable-on-affinity*:

assumes $m \neq 0$ *f integrable-on* (cbox a b)

shows $(\lambda x. f (m *_{\mathbb{R}} x + c))$ *integrable-on* $((\lambda x. (1 / m) *_{\mathbb{R}} x - ((1 / m) *_{\mathbb{R}} c))$ ‘cbox a b)

proof –

from *assms* **obtain** *I* **where** (*f has-integral I*) (cbox a b)

by (auto simp: integrable-on-def)

from *has-integral-affinity*[OF this *assms*(1), of *c*] **show** ?thesis

by (auto simp: integrable-on-def)

qed

lemma *has-integral-cmul-iff*:

assumes $c \neq 0$

shows $((\lambda x. c *_{\mathbb{R}} f x)$ *has-integral* $(c *_{\mathbb{R}} I))$ *A* \longleftrightarrow (*f has-integral I*) *A*

using *assms* *has-integral-cmul*[of *f I A c*]

has-integral-cmul[of $\lambda x. c *_{\mathbb{R}} f x$ *c* $*_{\mathbb{R}} I$ *A* *inverse c*] **by** (auto simp: field-simps)

lemma *has-integral-affinity'*:

fixes *a* :: 'a::euclidean-space

assumes (*f has-integral i*) (cbox a b) **and** $m > 0$

shows $((\lambda x. f(m *_{\mathbb{R}} x + c))$ *has-integral* $(i /_{\mathbb{R}} m \wedge DIM('a))$)
 (cbox $((a - c) /_{\mathbb{R}} m)$ $((b - c) /_{\mathbb{R}} m)$)

proof (cases cbox a b = {})

case *True*

hence (cbox $((a - c) /_{\mathbb{R}} m)$ $((b - c) /_{\mathbb{R}} m)$) = {}

using $\langle m > 0 \rangle$ **unfolding** *box-eq-empty* **by** (auto simp: algebra-simps)

with *True* **and** *assms* **show** ?thesis **by** *simp*

next

case *False*

have $((\lambda x. f (m *_{\mathbb{R}} x + c))$ *has-integral* $(1 / |m| \wedge DIM('a)) *_{\mathbb{R}} i$)
 $((\lambda x. (1 / m) *_{\mathbb{R}} x + - ((1 / m) *_{\mathbb{R}} c))$ ‘cbox a b)

using *assms* **by** (*intro has-integral-affinity*) *auto*

also have $((\lambda x. (1 / m) *_{\mathbb{R}} x + - ((1 / m) *_{\mathbb{R}} c))$ ‘cbox a b) =
 $((\lambda x. - ((1 / m) *_{\mathbb{R}} c) + x)$ ‘ $(\lambda x. (1 / m) *_{\mathbb{R}} x)$ ‘cbox a b)

by (*simp add: image-image algebra-simps*)

also have $(\lambda x. (1 / m) *_{\mathbb{R}} x)$ ‘cbox a b = cbox $((1 / m) *_{\mathbb{R}} a)$ $((1 / m) *_{\mathbb{R}} b)$

using $\langle m > 0 \rangle$ *False*

by (*subst image-smult-cbox*) *simp-all*

also have $(\lambda x. - ((1 / m) *_{\mathbb{R}} c) + x)$ ‘... = cbox $((a - c) /_{\mathbb{R}} m)$ $((b - c) /_{\mathbb{R}} m)$

by (*subst cbox-translation* [*symmetric*]) (*simp add: field-simps vector-add-divide-simps*)

finally show ?thesis **using** $\langle m > 0 \rangle$ **by** (*simp add: field-simps*)

qed

lemma *has-integral-affinity-iff*:

fixes *f* :: 'a :: euclidean-space \Rightarrow 'b :: real-normed-vector

assumes $m > 0$
shows $((\lambda x. f (m *_{\mathbb{R}} x + c)) \text{ has-integral } (I /_{\mathbb{R}} m \wedge \text{DIM}(a)))$
 $(\text{cbox } ((a - c) /_{\mathbb{R}} m) ((b - c) /_{\mathbb{R}} m)) \longleftrightarrow$
 $(f \text{ has-integral } I) (\text{cbox } a \ b) \text{ (is ?lhs = ?rhs)}$
proof
assume ?lhs
from *has-integral-affinity* [OF this, of $1 / m - c /_{\mathbb{R}} m$] **and** $\langle m > 0 \rangle$
show ?rhs **by** (*simp add: vector-add-divide-simps*) (*simp add: field-simps*)
next
assume ?rhs
from *has-integral-affinity* [OF this, of $m \ c$] **and** $\langle m > 0 \rangle$
show ?lhs **by** *simp*
qed

lemma *has-contour-integral-linepath-Reals-iff*:

fixes $a \ b :: \text{complex}$ **and** $f :: \text{complex} \Rightarrow \text{complex}$

assumes $a \in \text{Reals } b \in \text{Reals } \text{Re } a < \text{Re } b$

shows $(f \text{ has-contour-integral } I) (\text{linepath } a \ b) \longleftrightarrow$
 $((\lambda x. f (\text{of-real } x)) \text{ has-integral } I) \{\text{Re } a.. \text{Re } b\}$

proof –

from *assms* **have** [*simp*]: $\text{of-real } (\text{Re } a) = a \ \text{of-real } (\text{Re } b) = b$

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

from *assms* **have** $a \neq b$ **by** *auto*

have $((\lambda x. f (\text{of-real } x)) \text{ has-integral } I) (\text{cbox } (\text{Re } a) (\text{Re } b)) \longleftrightarrow$

$((\lambda x. f (a + b * \text{of-real } x - a * \text{of-real } x)) \text{ has-integral } I /_{\mathbb{R}} (\text{Re } b - \text{Re } a)) \{0..1\}$

by (*subst has-integral-affinity-iff* [*of Re b - Re a - Re a, symmetric*])

(*insert assms, simp-all add: field-simps scaleR-conv-of-real*)

also have $(\lambda x. f (a + b * \text{of-real } x - a * \text{of-real } x)) =$

$(\lambda x. (f (a + b * \text{of-real } x - a * \text{of-real } x) * (b - a)) /_{\mathbb{R}} (\text{Re } b - \text{Re } a))$

using $\langle a \neq b \rangle$ **by** (*auto simp: field-simps fun-eq-iff scaleR-conv-of-real*)

also have $(\dots \text{ has-integral } I /_{\mathbb{R}} (\text{Re } b - \text{Re } a)) \{0..1\} \longleftrightarrow$

$((\lambda x. f (\text{linepath } a \ b \ x) * (b - a)) \text{ has-integral } I) \{0..1\}$ **using** *assms*

by (*subst has-integral-cmul-iff*) (*auto simp: linepath-def scaleR-conv-of-real algebra-simps*)

also have $\dots \longleftrightarrow (f \text{ has-contour-integral } I) (\text{linepath } a \ b)$ **unfolding** *has-contour-integral-def*

by (*intro has-integral-cong*) (*simp add: vector-derivative-linepath-within*)

finally show ?thesis **by** *simp*

qed

lemma *contour-integrable-linepath-Reals-iff*:

fixes $a \ b :: \text{complex}$ **and** $f :: \text{complex} \Rightarrow \text{complex}$

assumes $a \in \text{Reals } b \in \text{Reals } \text{Re } a < \text{Re } b$

shows $(f \text{ contour-integrable-on linepath } a \ b) \longleftrightarrow$

$(\lambda x. f (\text{of-real } x)) \text{ integrable-on } \{\text{Re } a.. \text{Re } b\}$

using *has-contour-integral-linepath-Reals-iff* [OF *assms*, of *f*]

by (*auto simp: contour-integrable-on-def integrable-on-def*)

lemma *contour-integral-linepath-Reals-eq*:

fixes $a b :: \text{complex}$ **and** $f :: \text{complex} \Rightarrow \text{complex}$
assumes $a \in \text{Reals } b \in \text{Reals } \text{Re } a < \text{Re } b$
shows $\text{contour-integral } (\text{linepath } a \ b) \ f = \text{integral } \{ \text{Re } a .. \text{Re } b \} (\lambda x. f \ (\text{of-real } x))$
proof (*cases* f *contour-integrable-on* $\text{linepath } a \ b$)
case *True*
thus *?thesis* **using** *has-contour-integral-linepath-Reals-iff*[*OF assms, of f*]
using *has-contour-integral-integral has-contour-integral-unique* **by** *blast*
next
case *False*
thus *?thesis* **using** *contour-integrable-linepath-Reals-iff*[*OF assms, of f*]
by (*simp add: not-integrable-contour-integral not-integrable-integral*)
qed

lemma *has-contour-integral-linepath-same-Im-iff*:
fixes $a b :: \text{complex}$ **and** $f :: \text{complex} \Rightarrow \text{complex}$
assumes $\text{Im } a = \text{Im } b \ \text{Re } a < \text{Re } b$
shows $(f \ \text{has-contour-integral } I) \ (\text{linepath } a \ b) \ \longleftrightarrow$
 $((\lambda x. f \ (\text{of-real } x + \text{Im } a * i)) \ \text{has-integral } I) \ \{ \text{Re } a .. \text{Re } b \}$
proof –
have *deriv: vector-derivative* $((\lambda x. x - \text{Im } a * i) \circ \text{linepath } a \ b) \ (\text{at } y) = b - a$
for y
using *linepath-translate-right*[*of a - Im a * i b, symmetric*] **by** *simp*
have $(f \ \text{has-contour-integral } I) \ (\text{linepath } a \ b) \ \longleftrightarrow$
 $((\lambda x. f \ (x + \text{Im } a * i)) \ \text{has-contour-integral } I) \ (\text{linepath } (a - \text{Im } a * i) \ (b - \text{Im } a * i))$
using *linepath-translate-right*[*of a - Im a * i b*] *deriv* **by** (*simp add: has-contour-integral*)
also have $\dots \ \longleftrightarrow ((\lambda x. f \ (x + \text{Im } a * i)) \ \text{has-integral } I) \ \{ \text{Re } a .. \text{Re } b \}$ **using**
assms
by (*subst has-contour-integral-linepath-Reals-iff*) (*auto simp: complex-is-Real-iff*)
finally show *?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 $(f \ \text{contour-integrable-on } \text{linepath } a \ b) \ \longleftrightarrow$
 $(\lambda x. f \ (\text{of-real } x + \text{Im } a * i)) \ \text{integrable-on } \{ \text{Re } a .. \text{Re } b \}$
using *has-contour-integral-linepath-same-Im-iff*[*OF assms, of f*]
by (*auto simp: contour-integrable-on-def integrable-on-def*)

lemma *contour-integral-linepath-same-Im*:
fixes $a b :: \text{complex}$ **and** $f :: \text{complex} \Rightarrow \text{complex}$
assumes $\text{Im } a = \text{Im } b \ \text{Re } a < \text{Re } b$
shows $\text{contour-integral } (\text{linepath } a \ b) \ f = \text{integral } \{ \text{Re } a .. \text{Re } b \} (\lambda x. f \ (x + \text{Im } a * i))$
proof (*cases* f *contour-integrable-on* $\text{linepath } a \ b$)
case *True*
thus *?thesis* **using** *has-contour-integral-linepath-same-Im-iff*[*OF assms, of f*]

```

    using has-contour-integral-integral has-contour-integral-unique by blast
next
case False
thus ?thesis using contour-integrable-linepath-same-Im-iff[OF assms, of f]
  by (simp add: not-integrable-contour-integral not-integrable-integral)
qed

```

```

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

```

```

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

```

```

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

```

```

lemma open-contains-cbox:
  fixes x :: 'a :: euclidean-space
  assumes open A x ∈ A
  obtains a b where cbox a b ⊆ A x ∈ box a b ∀ i ∈ Basis. a · i < b · i
proof -
  from assms obtain R where R: R > 0 ball x R ⊆ A
  by (auto simp: open-contains-ball)
  define r :: real where r = R / (2 * sqrt DIM('a))
  from ⟨R > 0⟩ have [simp]: r > 0 by (auto simp: r-def)
  define d :: 'a where d = r *R Topology-Euclidean-Space.One
  have cbox (x - d) (x + d) ⊆ A
  proof safe
    fix y assume y: y ∈ cbox (x - d) (x + d)
    have dist x y = sqrt (∑ i ∈ Basis. (dist (x · i) (y · i))2)
    by (subst euclidean-dist-l2) (auto simp: L2-set-def)
    also from y have sqrt (∑ i ∈ Basis. (dist (x · i) (y · i))2) ≤ sqrt (∑ i ∈ (Basis::'a
set). r2)
    by (intro real-sqrt-le-mono sum-mono power-mono)
      (auto simp: dist-norm d-def cbox-def algebra-simps)
    also have ... = sqrt (DIM('a) * r2) by simp
    also have DIM('a) * r2 = (R / 2) ^ 2
    by (simp add: r-def power-divide)
    also have sqrt ... = R / 2
    using ⟨R > 0⟩ by simp
    also from ⟨R > 0⟩ have ... < R by simp

```

finally have $y \in \text{ball } x \ R$ **by** *simp*
with R **show** $y \in A$ **by** *blast*
qed
thus *?thesis*
using *that[$\text{of } x - d \ x + d$] by (auto simp: algebra-simps d-def box-def)*
qed

lemma *open-contains-box*:
fixes $x :: 'a :: \text{euclidean-space}$
assumes *open* A $x \in A$
obtains $a \ b$ **where** $\text{box } a \ b \subseteq A$ $x \in \text{box } a \ b$ $\forall i \in \text{Basis. } a \cdot i < b \cdot i$
proof –
from *assms* **obtain** $a \ b$ **where** $\text{cbox } a \ b \subseteq A$ $x \in \text{box } a \ b$ $\forall i \in \text{Basis. } a \cdot i < b \cdot i$
by (*rule open-contains-cbox*)
with *that[$\text{of } a \ b$] box-subset-cbox[$\text{of } a \ b$] show ?thesis* **by** *auto*
qed

lemma *analytic-onE-box*:
assumes f *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 *analytic-on* $\text{box } a \ b$
proof –
from *assms* **obtain** r **where** $r: r > 0$ f *holomorphic-on* $\text{ball } s \ r$
by (*auto simp: analytic-on-def*)
with *open-contains-box[$\text{of ball } s \ r \ s$] obtain* $a \ b$
where $\text{box } a \ b \subseteq \text{ball } s \ r$ $s \in \text{box } a \ b$ $\forall i \in \text{Basis. } a \cdot i < b \cdot i$ **by** *auto*
moreover from r **have** f *analytic-on* $\text{ball } s \ r$ **by** (*simp add: analytic-on-open*)
ultimately show *?thesis using that[$\text{of } a \ b$] analytic-on-subset[$\text{of - ball } s \ r \ \text{box } a \ b$]*
by (*auto simp: Basis-complex-def*)
qed

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

lemma *Re-image-box*:

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

lemma *Im-image-box*:

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

lemma *inner-image-cbox*:

assumes $(i :: 'a :: euclidean-space) \in Basis$
assumes $\forall i \in Basis. a \cdot i \leq b \cdot i$
shows $(\lambda x. x \cdot i)\ 'cbox\ a\ b = \{a \cdot i .. b \cdot i\}$

proof *safe*

fix x **assume** $x \in \{a \cdot i .. b \cdot i\}$
let $?y = (\sum_{j \in Basis. (if\ i = j\ then\ x\ else\ a \cdot j) *_{\mathbb{R}} j)$
from x **assms have** $?y \cdot i \in (\lambda x. x \cdot i)\ 'cbox\ a\ b$
by (*intro imageI*) (*auto simp: cbox-def*)
also have $?y \cdot i = (\sum_{j \in Basis. (if\ i = j\ then\ x\ else\ a \cdot j) * (j \cdot i))$
by (*simp add: inner-sum-left*)
also have $\dots = (\sum_{j \in Basis. if\ i = j\ then\ x\ else\ 0)$
by (*intro sum.cong*) (*auto simp: inner-not-same-Basis assms*)
also have $\dots = x$ **using** **assms by** *simp*
finally show $x \in (\lambda x. x \cdot i)\ 'cbox\ a\ b$.

qed (*insert assms, auto simp: cbox-def*)

lemma *Re-image-cbox*:

assumes $Re\ a \leq Re\ b\ Im\ a \leq Im\ b$
shows $Re\ 'cbox\ a\ b = \{Re\ a .. Re\ b\}$
using *inner-image-cbox*[of $1 :: complex\ a\ b$] **assms by** (*auto simp: Basis-complex-def*)

lemma *Im-image-cbox*:

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

lemma *analytic-onE-cball*:

assumes $f\ analytic\ on\ A\ s \in A\ ub > (0 :: real)$
obtains R **where** $R > 0\ R < ub$ $f\ analytic\ on\ cball\ s\ R$

proof –

from **assms obtain** r **where** $r > 0$ $f\ holomorphic\ on\ ball\ s\ r$
by (*auto simp: analytic-on-def*)
hence $f\ analytic\ on\ ball\ s\ r$ **by** (*simp add: analytic-on-open*)
hence $f\ analytic\ on\ cball\ s\ (\min\ (ub / 2)\ (r / 2))$
by (*rule analytic-on-subset, subst cball-subset-ball-iff*) (*use* $\langle r > 0 \rangle$ **in** *auto*)
moreover have $\min\ (ub / 2)\ (r / 2) > 0$ **and** $\min\ (ub / 2)\ (r / 2) < ub$
using $\langle r > 0 \rangle$ **and** $\langle ub > 0 \rangle$ **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 $(\lambda x. \text{pre-zeta } a (f x))$ 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 $(\bigwedge x. x \in A \implies f x \neq 1)$ $a > 0$

shows $(\lambda x. \text{hurwitz-zeta } a (f x))$ 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 $(\bigwedge x. x \in A \implies f x \neq 1)$

shows $(\lambda x. \text{zeta } (f x))$ 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*: $(\lambda s. \text{deriv zeta } s / \text{zeta } s)$ analytic-on $\{s. \text{Re } s \geq 1\} - \{1\}$

using *zeta-Re-ge-1-nonzero* **by** (*auto intro!: analytic-intros*)

lemma *cis-pi-half* [*simp*]: $\text{cis } (\pi / 2) = i$

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

lemma *mult-real-sqrt*: $x \geq 0 \implies x * \text{sqrt } y = \text{sqrt } (x^2 * y)$

by (*simp add: real-sqrt-mult*)

lemma *arcsin-pos*: $x \in \{0 <..1\} \implies \text{arcsin } x > 0$

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 $0 \in s$ f 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* $(\lambda n. \text{fds-nth } f n = \text{fds-nth } g n)$ *at-top* $s = s'$

shows *fds-converges* $f s \iff \text{fds-converges } g s'$

unfolding *fds-converges-def*

by (*intro summable-cong eventually-mono*[*OF* *assms*(1)]) (*simp-all add: assms*)

lemma *fds-abs-converges-cong*:

assumes *eventually* $(\lambda n. \text{fds-nth } f \ n = \text{fds-nth } g \ n)$ *at-top* $s = s'$

shows *fds-abs-converges* $f \ s \longleftrightarrow \text{fds-abs-converges } g \ s'$

unfolding *fds-abs-converges-def*

by (*intro summable-cong eventually-mono*[*OF assms(1)*]) (*simp-all add: assms*)

lemma *conv-abscissa-cong*:

assumes *eventually* $(\lambda n. \text{fds-nth } f \ n = \text{fds-nth } g \ n)$ *at-top*

shows *conv-abscissa* $f = \text{conv-abscissa } g$

proof –

have *fds-converges* $f = \text{fds-converges } g$

by (*intro ext fds-converges-cong assms refl*)

thus *?thesis* **by** (*simp add: conv-abscissa-def*)

qed

lemma *abs-conv-abscissa-cong*:

assumes *eventually* $(\lambda n. \text{fds-nth } f \ n = \text{fds-nth } g \ n)$ *at-top*

shows *abs-conv-abscissa* $f = \text{abs-conv-abscissa } g$

proof –

have *fds-abs-converges* $f = \text{fds-abs-converges } g$

by (*intro ext fds-abs-converges-cong assms refl*)

thus *?thesis* **by** (*simp add: abs-conv-abscissa-def*)

qed

definition *fds-remainder where*

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

lemma *fds-nth-remainder*: *fds-nth* $(\text{fds-remainder } m \ f) = (\lambda n. \text{if } n > m \ \text{then } \text{fds-nth } f \ n \ \text{else } 0)$

by (*simp add: fds-remainder-def fds-subseries-def fds-nth-fds'*)

lemma *fds-converges-remainder-iff* [*simp*]:

fds-converges $(\text{fds-remainder } m \ f) \ s \longleftrightarrow \text{fds-converges } f \ s$

by (*intro fds-converges-cong eventually-mono*[*OF eventually-gt-at-top*[*of m*]])

(*auto simp: fds-nth-remainder*)

lemma *fds-abs-converges-remainder-iff* [*simp*]:

fds-abs-converges $(\text{fds-remainder } m \ f) \ s \longleftrightarrow \text{fds-abs-converges } f \ s$

by (*intro fds-abs-converges-cong eventually-mono*[*OF eventually-gt-at-top*[*of m*]])

(*auto simp: fds-nth-remainder*)

lemma *fds-converges-remainder* [*intro*]:

fds-converges $f \ s \implies \text{fds-converges } (\text{fds-remainder } m \ f) \ s$

and *fds-abs-converges-remainder* [*intro*]:

fds-abs-converges $f \ s \implies \text{fds-abs-converges } (\text{fds-remainder } m \ f) \ s$

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*:
 $eval-fds (fds-remainder\ m\ f)\ s = (\sum n. fds-nth\ f\ (n + Suc\ m) / nat-power\ (n + Suc\ m)\ s)$
(is - = *suminf* ($\lambda n. ?f\ (n + Suc\ m)$))
proof (*cases* *fds-converges* *f* *s*)
case *False*
hence $\neg fds-converges (fds-remainder\ m\ f)\ s$ **by** *simp*
hence ($\lambda x. (\lambda n. fds-nth (fds-remainder\ m\ f)\ n / nat-power\ n\ s)\ sums\ x$) = ($\lambda -. False$)
by (auto simp: *fds-converges-def* *summable-def*)
hence $eval-fds (fds-remainder\ m\ f)\ s = (THE\ -. False)$
by (*simp* *add: eval-fds-def suminf-def*)
moreover from *False* **have** $\neg summable (\lambda n. ?f\ (n + Suc\ m))$ **unfolding**
fds-converges-def
by (*subst* *summable-iff-shift*) *auto*
hence ($\lambda x. (\lambda n. ?f\ (n + Suc\ m))\ sums\ x$) = ($\lambda -. False$)
by (auto simp: *summable-def*)
hence $suminf (\lambda n. ?f\ (n + Suc\ m)) = (THE\ -. False)$
by (*simp* *add: suminf-def*)
ultimately show *?thesis* **by** *simp*

next
case *True*
hence *: *fds-converges* (*fds-remainder* *m* *f*) *s* **by** *simp*
have $eval-fds (fds-remainder\ m\ f)\ s = (\sum n. fds-nth (fds-remainder\ m\ f)\ n / nat-power\ n\ s)$
unfolding *eval-fds-def* ..
also have $\dots = (\sum n. fds-nth (fds-remainder\ m\ f)\ (n + Suc\ m) / nat-power\ (n + Suc\ m)\ s)$
using * **unfolding** *fds-converges-def*
by (*subst* *suminf-minus-initial-segment*) (auto simp: *fds-nth-remainder*)
also have ($\lambda n. fds-nth (fds-remainder\ m\ f)\ (n + Suc\ m) = \lambda n. fds-nth\ f\ (n + Suc\ m)$)
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* $A \wedge x. x \in A \implies \text{Re } (g x) > \text{conv-abscissa } f$
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* $A \wedge x. x \in A \implies \text{Re } (g x) > \text{conv-abscissa } f$
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*] *assms(2)*
by (*auto simp: o-def*)

lemma *homotopic-loopsI*:
fixes $h :: \text{real} \times \text{real} \Rightarrow -$
assumes *continuous-on* $(\{0..1\} \times \{0..1\})$ h
 $h \text{ ' } (\{0..1\} \times \{0..1\}) \subseteq s$
 $\wedge x. x \in \{0..1\} \implies h (0, x) = p x$
 $\wedge x. x \in \{0..1\} \implies h (1, x) = q x$
 $\wedge x. x \in \{0..1\} \implies \text{pathfinish } (h \circ \text{Pair } x) = \text{pathstart } (h \circ \text{Pair } x)$
shows *homotopic-loops* $s p q$
using *assms* **unfolding** *homotopic-loops* **by** (*intro exI*[*of - h*]) *auto*

lemma *continuous-on-linepath* [*continuous-intros*]:
assumes *continuous-on* A a *continuous-on* A b *continuous-on* A f
shows *continuous-on* A $(\lambda x. \text{linepath } (a x) (b x) (f x))$
using *assms* **by** (*auto simp: linepath-def intro!*; *continuous-intros assms*)

lemma *continuous-on-part-circlepath* [*continuous-intros*]:
assumes *continuous-on* A c *continuous-on* A r *continuous-on* A a *continuous-on* A b
 $\text{continuous-on } A f$
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**
 $b1 = a1 + 2 * \text{of-int } k * \text{pi}$ **and** $b2 = a2 + 2 * \text{of-int } k * \text{pi}$
shows *homotopic-loops* A $(\text{part-circlepath } c r a1 b1)$ $(\text{part-circlepath } c r a2 b2)$
proof –
define h **where** $h = (\lambda(x,y). \text{part-circlepath } c r (\text{linepath } a1 a2 x) (\text{linepath } b1 b2 x) y)$
show *?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  $h \text{ ' } (\{0..1\} \times \{0..1\}) \subseteq \text{sphere } c \ r$ 
  by (auto simp: h-def part-circlepath-def dist-norm norm-mult)
also have  $\dots \subseteq A$  by fact
finally show  $h \text{ ' } (\{0..1\} \times \{0..1\}) \subseteq A .$ 
next
  fix  $x :: \text{real}$  assume  $x \in \{0..1\}$ 
  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
    cis-mult [symmetric] cis-divide [symmetric] assms)
qed
qed

```

```

lemma homotopic-pathsI:
  fixes  $h :: \text{real} \times \text{real} \Rightarrow -$ 
  assumes continuous-on  $(\{0..1\} \times \{0..1\}) \ h$ 
  assumes  $h \text{ ' } (\{0..1\} \times \{0..1\}) \subseteq s$ 
  assumes  $\bigwedge x. x \in \{0..1\} \implies h \ (0, x) = p \ x$ 
  assumes  $\bigwedge x. x \in \{0..1\} \implies h \ (1, x) = q \ x$ 
  assumes  $\bigwedge x. x \in \{0..1\} \implies \text{pathstart } (h \circ \text{Pair } x) = \text{pathstart } p$ 
  assumes  $\bigwedge x. x \in \{0..1\} \implies \text{pathfinish } (h \circ \text{Pair } x) = \text{pathfinish } p$ 
  shows homotopic-paths  $s \ p \ q$ 
  using assms unfolding homotopic-paths by (intro exI[of - h]) auto

```

```

lemma part-circlepath-conv-subpath:
   $\text{part-circlepath } c \ r \ a \ b = \text{subpath } (a / (2*\pi)) \ (b / (2*\pi)) \ (\text{circlepath } c \ r)$ 
  by (simp add: part-circlepath-def circlepath-def subpath-def linepath-def alge-
bra-simps exp-eq-polar)

```

```

lemma homotopic-paths-part-circlepath:
  assumes  $a \leq b \ b \leq c$ 
  assumes  $\text{path-image } (\text{part-circlepath } C \ r \ a \ c) \subseteq A \ r \geq 0$ 
  shows homotopic-paths  $A \ (\text{part-circlepath } C \ r \ a \ c)$ 
   $(\text{part-circlepath } C \ r \ a \ b \ +++ \ \text{part-circlepath } C \ r \ b \ c)$ 
  (is homotopic-paths - ?g (?h1 +++ ?h2))
proof (cases a = c)
  case False
  with assms have  $a < c$  by simp
  define slope where  $\text{slope} = (b - a) / (c - a)$ 
  from assms and  $\langle a < c \rangle$  have  $\text{slope} : \text{slope} \in \{0..1\}$ 
  by (auto simp: field-simps slope-def)
  define  $f :: \text{real} \Rightarrow \text{real}$  where
     $f = \text{linepath } 0 \ \text{slope} \ +++ \ \text{linepath } \text{slope} \ 1$ 

  show ?thesis

```

```

proof (rule homotopic-paths-reparametrize)
  fix  $t :: \text{real}$  assume  $t: t \in \{0..1\}$ 
  show ( $?h1$   $+++$   $?h2$ )  $t = ?g (f t)$ 
  proof (cases  $t \leq 1 / 2$ )
    case True
      hence  $?g (f t) = C + r * \text{cis} ((1 - f t) * a + f t * c)$ 
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      also from True  $\langle a < c \rangle$  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
also from True have  $C + r * \text{cis} \dots = (?h1$   $+++$   $?h2)$   $t$ 
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      finally show  $?thesis$  ..
    next
      case False
      hence  $?g (f t) = C + r * \text{cis} ((1 - f t) * a + f t * c)$ 
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      also from False  $\langle a < c \rangle$  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
also from False have  $C + r * \text{cis} \dots = (?h1$   $+++$   $?h2)$   $t$ 
      by (simp add: joinpaths-def part-circlepath-def exp-eq-polar linepath-def)
      finally show  $?thesis$  ..
    qed
  next
    from slope have  $\text{path-image } f \subseteq \{0..1\}$ 
    by (auto simp:  $f$ -def path-image-join closed-segment-eq-real-ivl)
    thus  $f ' \{0..1\} \subseteq \{0..1\}$  by (simp add: path-image-def)
  next
    have  $\text{path } f$  unfolding  $f$ -def by auto
    thus continuous-on  $\{0..1\}$   $f$  by (simp add: path-def)
  qed (insert assms, auto simp:  $f$ -def joinpaths-def linepath-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: fun-eq-iff joinpaths-def part-circlepath-def)
  thus  $?thesis$  using assms by simp
qed

lemma has-contour-integral-mirror-iff:
  assumes valid-path  $g$ 

```

shows $(f \text{ has-contour-integral } I) (-g) \longleftrightarrow ((\lambda x. -f (-x)) \text{ has-contour-integral } I) g$
proof –
from *assms* **have** $g \text{ piecewise-differentiable-on } \{0..1\}$
by (*auto simp: valid-path-def piecewise-C1-imp-differentiable*)
then obtain S **where** $S: \text{finite } S \wedge x. x \in \{0..1\} - S \implies g \text{ differentiable at } x \text{ within } \{0..1\}$
unfolding *piecewise-differentiable-on-def* **by** *blast*
have $S': g \text{ differentiable at } x \text{ if } x \in \{0..1\} - (\{0, 1\} \cup S) \text{ for } x$
proof –
from *that* **have** $x \in \text{interior } \{0..1\}$ **by** *auto*
with $S(2)[\text{of } x]$ **that show** *?thesis* **by** (*auto simp: at-within-interior[of - {0..1}]*)
qed

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

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

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

lemma *contour-integrable-neg-iff*:
 $(\lambda x. -f x)$ *contour-integrable-on* $g \iff f$ *contour-integrable-on* g
using *contour-integrable-neg*[of f g] *contour-integrable-neg*[of $\lambda x. -f x$ g] **by** *auto*

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

lemma *minus-cis*: $-cis\ x = cis\ (x + \pi)$
by (*simp add: complex-eq-iff*)

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

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

lemma *path-mirror* [*intro*]: $path\ (g :: - \Rightarrow 'b::topological-group-add) \implies path\ (-g)$
by (*auto simp: path-def intro!: continuous-intros*)

lemma *path-mirror-iff* [*simp*]: $\text{path } (-g :: - \Rightarrow 'b::\text{topological-group-add}) \longleftrightarrow \text{path } g$
using *path-mirror*[*of g*] *path-mirror*[*of -g*] **by** (*auto simp: fun-Compl-def*)

lemma *valid-path-mirror* [*intro*]: $\text{valid-path } g \Longrightarrow \text{valid-path } (-g)$
by (*auto simp: valid-path-def fun-Compl-def piecewise-C1-differentiable-neg*)

lemma *valid-path-mirror-iff* [*simp*]: $\text{valid-path } (-g) \longleftrightarrow \text{valid-path } g$
using *valid-path-mirror*[*of g*] *valid-path-mirror*[*of -g*] **by** (*auto simp: fun-Compl-def*)

lemma *pathstart-mirror* [*simp*]: $\text{pathstart } (-g) = -\text{pathstart } g$
and *pathfinish-mirror* [*simp*]: $\text{pathfinish } (-g) = -\text{pathfinish } g$
by (*simp-all add: pathstart-def pathfinish-def*)

lemma *path-image-mirror*: $\text{path-image } (-g) = \text{uminus } \text{'path-image } g$
by (*auto simp: path-image-def*)

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

lemma *contour-integral-spike-finite-simple-path*:
assumes *finite A simple-path g g = g' $\wedge x. x \in \text{path-image } g - A \Longrightarrow f \ x = f' \ x$*
shows $\text{contour-integral } g \ f = \text{contour-integral } g' \ f'$
unfolding *contour-integral-integral*
proof (*rule integral-spike*)
have *finite* ($g - \text{'A} \cap \{0 < .. < 1\}$) **using** $\langle \text{simple-path } g \rangle \langle \text{finite } A \rangle$
by (*intro finite-vimage-IntI simple-path-inj-on*) *auto*

hence *finite* $(\{0, 1\} \cup g - 'A \cap \{0 < .. < 1\})$ **by** *auto*
thus *negligible* $(\{0, 1\} \cup g - 'A \cap \{0 < .. < 1\})$ **by** *(rule negligible-finite)*
next
fix x **assume** $x \in \{0..1\} - (\{0, 1\} \cup g - 'A \cap \{0 < .. < 1\})$
hence $g x \in \text{path-image } g - A$ **by** *(auto simp: path-image-def)*
from *assms(4)[OF this]* **and** *assms(3)*
show $f' (g' x) * \text{vector-derivative } g' (\text{at } x) = f (g x) * \text{vector-derivative } g (\text{at } x)$ **by** *simp*
qed

proposition *contour-integral-bound-part-circlepath-strong*:
assumes fi : f *contour-integrable-on part-circlepath* $z r s t$
and *finite* k **and** le : $0 \leq B$ $0 < r s \leq t$
and B : $\bigwedge x. x \in \text{path-image}(\text{part-circlepath } z r s t) - k \implies \text{norm}(f x) \leq B$
shows $cmod (\text{contour-integral } (\text{part-circlepath } z r s t) f) \leq B * r * (t - s)$
proof –
from fi **have** $(f \text{ has-contour-integral } \text{contour-integral } (\text{part-circlepath } z r s t) f)$
 $(\text{part-circlepath } z r s t)$
by *(rule has-contour-integral-integral)*
from *has-contour-integral-bound-part-circlepath-strong[OF this assms(2-)]* **show**
?thesis **by** *auto*
qed

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

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

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

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

lemma *Union-atLeastAtMost-real-of-nat*:
assumes $a < b$

```

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

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

```

```

lemma nat-sum-has-integral-ceiling:
  fixes  $f :: nat \Rightarrow 'a :: banach$ 
  assumes  $mn: m < n$ 
  shows  $((\lambda x. f\ (nat\ \lceil x \rceil))\ has\_integral\ sum\ f\ \{m<..n\})\ \{real\ m..real\ n\}$ 

```


proof –
define D **where** $D = (\lambda i. \{ \text{real } i.. \text{real } (\text{Suc } i) \}) \text{ ‘ } \{ m..<n \}$
have D : D *division-of* $\{ m..n \}$
using *Union-atLeastAtMost-real-of-nat*[$OF\ mn$] **by** (*simp add: division-of-def*
D-def)
have $((\lambda x. f (\text{nat } \lceil x \rceil)) \text{ has-integral } (\sum X \in D. f (\text{nat } \lfloor \text{Sup } X \rfloor))) \{ \text{real } m.. \text{real } n \}$
proof (*rule has-integral-combine-division*)
fix X **assume** X : $X \in D$
have $\text{nat } \lceil x \rceil = \text{nat } \lfloor \text{Sup } X \rfloor$ **if** $x \in X - \{ \text{Inf } X \}$ **for** x
using *that X* **by** (*auto simp: D-def nat-eq-iff ceiling-eq-iff*)
hence $((\lambda x. f (\text{nat } \lceil x \rceil)) \text{ has-integral } f (\text{nat } \lfloor \text{Sup } X \rfloor)) X \longleftrightarrow$
 $((\lambda x. f (\text{nat } \lfloor \text{Sup } X \rfloor)) \text{ has-integral } f (\text{nat } \lfloor \text{Sup } X \rfloor)) X$ **using** X
by (*intro has-integral-spike-eq*[*of* $\{ \text{Inf } X \}$]) *auto*
also from X **have** ... **using** *has-integral-const-real*[*of* $f (\text{nat } \lfloor \text{Sup } X \rfloor)$] *Inf X*
Sup X]
by (*auto simp: D-def*)
finally show $((\lambda x. f (\text{nat } \lceil x \rceil)) \text{ has-integral } f (\text{nat } \lfloor \text{Sup } X \rfloor)) X$.
qed fact+
also have $(\sum X \in D. f (\text{nat } \lfloor \text{Sup } X \rfloor)) = (\sum k \in \{ m..<n \}. f (\text{Suc } k))$
unfolding *D-def* **by** (*subst sum.reindex*) (*auto simp: inj-on-def nat-add-distrib*)
also have ... $= (\sum k \in \{ m <.. n \}. f k)$
by (*intro sum.reindex-bij-witness*[*of* $\lambda x. x - 1$] *Suc*) *auto*
finally show *?thesis* .
qed

lemma *zeta-partial-sum-le*:
fixes $x :: \text{real}$ **and** $m :: \text{nat}$
assumes x : $x \in \{ 0 <.. 1 \}$
shows $(\sum k=1..m. \text{real } k \text{ powr } (x - 1)) \leq \text{real } m \text{ powr } x / x$
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 $(\sum k \in \dots \text{real } k \text{ powr } (x - 1)) = 1 + (\sum k \in \{ 1 <..m \}. \text{real } k \text{ powr } (x - 1))$
 $(x - 1)$
by simp
also have $(\sum k \in \{ 1 <..m \}. \text{real } k \text{ powr } (x - 1)) \leq \text{real } m \text{ powr } x / x - 1 / x$
proof (*rule has-integral-le*)
show $((\lambda t. (\text{nat } \lceil t \rceil) \text{ powr } (x - 1)) \text{ has-integral } (\sum n \in \{ 1 <..m \}. n \text{ powr } (x - 1))) \{ \text{real } 1..m \}$
using m **by** (*intro nat-sum-has-integral-ceiling*) *auto*
next
have $((\lambda t. t \text{ powr } (x - 1)) \text{ has-integral } (\text{real } m \text{ powr } x / x - \text{real } 1 \text{ powr } x / x))$
 $\{ \text{real } 1.. \text{real } m \}$
by (*intro fundamental-theorem-of-calculus*)
(insert x m, auto simp flip: has-field-derivative-iff-has-vector-derivative

```

      intro!: derivative-eq-intros)
    thus ((λt. t powr (x - 1)) has-integral (real m powr x / x - 1 / x)) {real
1..real m}
      by simp
    qed (insert x, auto intro!: powr-mono2)
    also have 1 + (real m powr x / x - 1 / x) ≤ real m 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: x > 0 and m: m > 0
  shows (∑ n=1..m. real n powr (x - 1)) ≤ m powr x * (1 / x + 1 / m)
proof (cases x > 1)
  case False
  with assms have (∑ n=1..m. real n powr (x - 1)) ≤ m powr x / x
    by (intro zeta-partial-sum-le) auto
  also have ... ≤ m powr x * (1 / x + 1 / m)
    using assms by (simp add: field-simps)
  finally show ?thesis .
next
  case True
  have (∑ n∈{1..m}. n powr (x - 1)) = (∑ n∈insert m {0..<m}. n powr (x -
1))
    by (intro sum.mono-neutral-left) auto
  also have ... = m powr (x - 1) + (∑ n∈{0..<m}. n powr (x - 1)) by simp
  also have (∑ n∈{0..<m}. n powr (x - 1)) ≤ real m powr x / x
  proof (rule has-integral-le)
    show ((λt. (nat ⌊t⌋) powr (x - 1)) has-integral (∑ n∈{0..<m}. n powr (x -
1))) {real 0..m}
      using m by (intro nat-sum-has-integral-floor) auto
  next
    show ((λt. t powr (x - 1)) has-integral (real m powr x / x)) {real 0..real m}
      using has-integral-powr-from-0[of x - 1] x by auto
  next
    fix t assume t ∈ {real 0..real m}
    with ⟨x > 1⟩ show real (nat ⌊t⌋) powr (x - 1) ≤ t powr (x - 1)
      by (cases t = 0) (auto intro: powr-mono2)
  qed
  also have m powr (x - 1) + m powr x / x = m 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 :: nat ⇒ -) ∈ O(λ-. 1)
  obtains C where C ≥ lb ∧ n. norm (f n) ≤ 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 *: $\text{norm } (f \ n) \leq \text{Max } (\{C, lb\} \cup (\text{norm } 'f' \ \{..<N\}))$ **for** n
by (*cases* $n \geq N$) (*subst Max-ge-iff; force simp: image-iff*) +
moreover **have** $\text{Max } (\{C, lb\} \cup (\text{norm } 'f' \ \{..<N\})) \geq lb$
by (*intro Max.coboundedI*) *auto*
ultimately show *?thesis* **using** *that* **by** *blast*

qed

lemma *natfun-bigo-iff-Bseq*: $f \in O(\lambda-. \ 1) \longleftrightarrow Bseq \ f$

proof

assume *Bseq f*
then **obtain** C **where** $C > 0 \ \bigwedge n. \text{norm } (f \ n) \leq C$ **by** (*auto simp: Bseq-def*)
thus $f \in O(\lambda-. \ 1)$ **by** (*intro bigoI[of - C]*) *auto*

next

assume $f \in O(\lambda-. \ 1)$
from *natfun-bigo-1E* [*OF this, where lb = 1*] **obtain** C **where** $C \geq 1 \ \bigwedge n. \text{norm } (f \ n) \leq C$
by *auto*
thus *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*: $\bigwedge x \ y. 0 \leq x \implies x \leq y \implies f \ y \leq f \ x$
shows $(\sum n. f \ (\text{real } (\text{Suc } n))) \leq \text{set-nn-integral } \text{lborel } \{0..\} \ f$

proof –

have $(\sum n. f \ (\text{real } (\text{Suc } n))) =$
 $(\sum n. \int^{+x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}}. (f \ (\text{real } (\text{Suc } n))) \ \partial \text{lborel})$
by (*subst nn-integral-cmult-indicator*) *auto*
also **have** $\text{nat } \lceil x \rceil = \text{Suc } n$ **if** $x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}$ **for** $x \ n$
using *that* **by** (*auto simp: nat-eq-iff ceiling-eq-iff*)
hence $(\sum n. \int^{+x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}}. (f \ (\text{real } (\text{Suc } n))) \ \partial \text{lborel}) =$
 $(\sum n. \int^{+x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}}. (f \ (\text{real } (\text{nat } \lceil x \rceil))) \ \partial \text{lborel})$
by (*intro suminf-cong nn-integral-cong*) (*auto simp: indicator-def*)
also **have** $\dots = (\int^{+x \in (\bigcup i. \{\text{real } i <.. \text{real } (\text{Suc } i)\}}). (f \ (\text{nat } \lceil x :: \text{real} \rceil)) \ \partial \text{lborel})$
by (*subst nn-integral-disjoint-family*)
(*auto simp: disjoint-family-on-def*)
also **have** $\dots \leq (\int^{+x \in \{0..\}}. (f \ x) \ \partial \text{lborel})$
by (*intro nn-integral-mono*) (*auto simp: indicator-def intro!: decreasing*)
finally **show** *?thesis* .

qed

lemma *nn-integral-has-integral-lebesgue*:

fixes $f :: 'a :: \text{euclidean-space} \Rightarrow \text{real}$
assumes *nonneg*: $\bigwedge x. x \in \Omega \implies 0 \leq f \ x$ **and** *I*: (*f has-integral I*) Ω
shows $\text{integral}^N \ \text{lborel } (\lambda x. \text{indicator } \Omega \ x * f \ x) = I$

proof –

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

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

qed

lemma *decreasing-sum-le-integral*:

fixes $f :: \text{real} \Rightarrow \text{real}$
assumes nonneg : $\bigwedge x. x \geq 0 \implies f x \geq 0$
assumes decreasing : $\bigwedge x y. 0 \leq x \implies x \leq y \implies f y \leq f x$
assumes integral : $(f \text{ has-integral } I) \{0..\}$
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 $(\text{intro has-integral-nonneg}[OF \text{integral}] \text{ nonneg}) \text{ auto}$
have $(\sum n. \text{ennreal } (f (\text{Suc } n))) =$
 $(\sum n. \int^{+x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}}. \text{ennreal } (f (\text{Suc } n)) \partial \text{lborel})$
by $(\text{subst nn-integral-cmult-indicator}) \text{ auto}$
also have $\text{nat } \lceil x \rceil = \text{Suc } n$ **if** $x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}$ **for** $x n$
using that **by** $(\text{auto simp: nat-eq-iff ceiling-eq-iff})$
hence $(\sum n. \int^{+x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}}. \text{ennreal } (f (\text{Suc } n)) \partial \text{lborel}) =$
 $(\sum n. \int^{+x \in \{\text{real } n <.. \text{real } (\text{Suc } n)\}}. \text{ennreal } (f (\text{real } (\text{nat } \lceil x \rceil))) \partial \text{lborel})$
by $(\text{intro suminf-cong nn-integral-cong}) (\text{auto simp: indicator-def})$
also have $\dots = (\int^{+x \in (\bigcup i. \{\text{real } i <.. \text{real } (\text{Suc } i)\}}). \text{ennreal } (f (\text{nat } \lceil x :: \text{real} \rceil))) \partial \text{lborel}$
by $(\text{subst nn-integral-disjoint-family})$
 $(\text{auto simp: disjoint-family-on-def intro!: measurable-completion})$
also have $\dots \leq (\int^{+x \in \{0..\}}. \text{ennreal } (f x) \partial \text{lborel})$
by $(\text{intro nn-integral-mono}) (\text{auto simp: indicator-def nonneg intro!: decreasing})$
also have $\dots = (\int^{+x. \text{ennreal } (\text{indicat-real } \{0..\} x *_R f x) \partial \text{lborel})$
by $(\text{intro nn-integral-cong}) (\text{auto simp: indicator-def})$

also have $\dots = \text{ennreal } I$
using *nn-integral-has-integral-lebesgue*[*OF nonneg integral*] **by** (*auto simp: non-neg*)
finally have $*$: $(\sum n. \text{ennreal } (f \text{ (Suc } n))) \leq \text{ennreal } I$.
from $*$ **show** *summable: summable* $(\lambda i. f \text{ (real (Suc } i)))$
by (*intro summable-suminf-not-top*) (*auto simp: top-unique intro: nonneg*)
note $*$
also from *summable* **have** $(\sum n. \text{ennreal } (f \text{ (Suc } n))) = \text{ennreal } (\sum n. f \text{ (Suc } n))$
by (*subst suminf-ennreal2*) (*auto simp: o-def nonneg*)
finally show $(\sum n. f \text{ (real (Suc } n))) \leq I$ **by** (*subst (asm) ennreal-le-iff*) *auto*
qed

lemma *decreasing-sum-le-integral'*:

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

lemma *norm-suminf-le*:

assumes $\bigwedge n. \text{norm } (f n :: 'a :: \text{banach}) \leq g n$ *summable* g
shows *norm* (*suminf* f) \leq *suminf* g
proof –
have $*$: *summable* $(\lambda n. \text{norm } (f n))$ **using** *assms*
by (*intro summable-norm summable-comparison-test*[*OF - assms(2)*] *exI[of - 0]*) *auto*
hence *norm* (*suminf* f) \leq $(\sum n. \text{norm } (f n))$ **by** (*intro summable-norm*) *auto*
also have $\dots \leq$ *suminf* g **by** (*intro suminf-le * assms allI*)
finally show *?thesis* .
qed

lemma *of-nat-powr-neq-1-complex* [*simp*]:

assumes $n > 1$ $\text{Re } s \neq 0$
shows *of-nat* n *powr* $s \neq (1 :: \text{complex})$
proof –
have *norm* (*of-nat* n *powr* s) $=$ *real* n *powr* $\text{Re } s$
by (*simp add: norm-powr-real-powr*)
also have $\dots \neq 1$
using *assms* **by** (*auto simp: powr-def*)
finally show *?thesis* **by** *auto*

qed

lemma *abs-summable-on-uminus-iff*:

$(\lambda x. -f x)$ *abs-summable-on* $A \longleftrightarrow f$ *abs-summable-on* A
using *abs-summable-on-uminus*[of f A] *abs-summable-on-uminus*[of $\lambda x. -f x$ A]
by *auto*

lemma *abs-summable-on-cmult-right-iff*:

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

lemma *abs-summable-on-cmult-left-iff*:

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

lemma *fds-logderiv-completely-multiplicative*:

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

qed

lemma *fds-nth-logderiv-completely-multiplicative*:

fixes $f :: 'a :: \{\text{real-normed-field}\}$ *fds*
assumes *completely-multiplicative-function* (*fds-nth* f) *fds-nth* f $1 \neq 0$
shows *fds-nth* (*fds-deriv* f / f) $n = -$ *fds-nth* $f \ n * \text{mangoldt } n$
using *assms* **by** (*subst fds-logderiv-completely-multiplicative*) (*simp-all add: fds-nth-fds'*)

lemma *eval-fds-logderiv-completely-multiplicative*:

fixes $s :: 'a :: \text{dirichlet-series}$ **and** $l :: 'a$ **and** $f :: 'a$ *fds*
defines $h \equiv \text{fds-deriv } f / f$
assumes *completely-multiplicative-function* (*fds-nth* f) **and** [*simp*]: *fds-nth* f $1 \neq 0$
assumes $s \cdot 1 > \text{abs-conv-abscissa } f$
shows $(\lambda p. \text{of-real } (\ln (\text{real } p)) * (1 / (1 - \text{fds-nth } f \ p / \text{nat-power } p \ s) - 1))$
abs-summable-on $\{p. \text{prime } p\}$ (**is** *?th1*)

and $eval\text{-}fds\ h\ s = -(\sum_{ap \mid prime\ p} of\text{-}real\ (ln\ (real\ p)) * (1 / (1 - fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) - 1))$ (**is** *?th2*)

proof –

let $?P = \{p::nat.\ prime\ p\}$

interpret f : *completely-multiplicative-function* $fds\text{-}nth\ f$ **by** *fact*

have $fds\text{-}abs\text{-}converges\ h\ s$

using $abs\text{-}conv\text{-}abscissa\text{-}completely\text{-}multiplicative\text{-}log\text{-}deriv[OF\ assms(2)]\ assms$

by (*intro* $fds\text{-}abs\text{-}converges$) *auto*

hence $*$: $(\lambda n.\ fds\text{-}nth\ h\ n / nat\text{-}power\ n\ s)$ *abs-summable-on UNIV*

by (*auto simp: h-def fds-abs-converges-altdef'*)

note $*$

also have $(\lambda n.\ fds\text{-}nth\ h\ n / nat\text{-}power\ n\ s)$ *abs-summable-on UNIV* \longleftrightarrow

$(\lambda x.\ -fds\text{-}nth\ f\ x * mangoldt\ x / nat\text{-}power\ x\ s)$ *abs-summable-on Collect primepow*

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

by (*intro* $abs\text{-}summable\text{-}on\text{-}cong\text{-}neutral$) (*auto simp: fds-nth-fds mangoldt-def*)

finally have $sum1$: $(\lambda x.\ -fds\text{-}nth\ f\ x * mangoldt\ x / nat\text{-}power\ x\ s)$

$abs\text{-}summable\text{-}on\ Collect\ primepow$

by (*rule* $abs\text{-}summable\text{-}on\text{-}subset$) *auto*

also have $?this \longleftrightarrow (\lambda(p,k).\ -fds\text{-}nth\ f\ (p \wedge Suc\ k) * mangoldt\ (p \wedge Suc\ k) / nat\text{-}power\ (p \wedge Suc\ k)\ s)$ *abs-summable-on (?P × UNIV)*

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

by (*intro* $abs\text{-}summable\text{-}on\text{-}reindex\text{-}bij\text{-}betw$ [*symmetric*])

also have $\dots \longleftrightarrow (\lambda(p,k).\ -((fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ k * of\text{-}real\ (ln\ (real\ p))))$

$abs\text{-}summable\text{-}on\ (?P \times UNIV)$

unfolding $case\text{-}prod\text{-}unfold$

by (*intro* $abs\text{-}summable\text{-}on\text{-}cong$, $subst\ mangoldt\text{-}primepow$)

$(auto\ simp: f.\text{mult}\ f.\text{power}\ nat\text{-}power\text{-}mult\text{-}distrib\ nat\text{-}power\text{-}power\text{-}left\ power\text{-}divide\ dest: prime\text{-}gt\text{-}1\text{-}nat)$

finally have $sum2$: \dots .

have $sum4$: $summable\ (\lambda n.\ (norm\ (fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s)) \wedge Suc\ n)$ **if** p :

$prime\ p$ **for** p

proof –

have $summable\ (\lambda n.\ |ln\ (real\ p)| * (norm\ (fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s)) \wedge Suc\ n)$

n

using p *abs-summable-on-Sigma-project2[OF sum2, of p]* **unfolding** $abs\text{-}summable\text{-}on\text{-}nat\text{-}iff'$

by (*simp add: norm-power norm-mult norm-divide mult-ac del: power-Suc*)

thus $?thesis$ **by** (*rule* $summable\text{-}mult\text{-}D$) (*insert* p , *auto dest: prime-gt-1-nat*)

qed

have $sums$: $(\lambda n.\ (fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ n)$ $sums$

$(1 / (1 - fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) - 1)$ **if** p : $prime\ p$ **for** $p :: nat$

proof –

from $sum4[OF\ p]$ **have** $norm\ (fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) < 1$

unfolding $summable\text{-}Suc\text{-}iff$ **by** (*simp add: summable-geometric-iff*)

from $geometric\text{-}sums[OF\ this]$ **show** $?thesis$ **by** (*subst* $sums\text{-}Suc\text{-}iff$) *auto*

qed

have eq: $(\sum_a k. - ((fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ k * of\text{-}real\ (ln\ (real\ p))))$
 $=$
 $-(of\text{-}real\ (ln\ (real\ p)) * (1 / (1 - fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) - 1))$
if p : *prime* p **for** p
proof -
have $(\sum_a k. - ((fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ k * of\text{-}real\ (ln\ (real\ p)))) =$
 $(\sum_a k. (fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ k) * of\text{-}real\ (-ln\ (real\ p))$
using *sum4*[*of* p] p
by (*subst infsetsum-cmult-left* [*symmetric*])
(auto simp: abs-summable-on-nat-iff' norm-power simp del: power-Suc)
also have $(\sum_a k. (fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ k) =$
 $(1 / (1 - fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) - 1)$ **using** *sum4*[*OF* p]
sums[*OF* p]
by (*subst infsetsum-nat'*)
(auto simp: sums-iff abs-summable-on-nat-iff' norm-power simp del:
power-Suc)
finally show *?thesis* **by** (*simp add: mult-ac*)
qed

have *sum3*: $(\lambda x. \sum_a y. - ((fds\text{-}nth\ f\ x / nat\text{-}power\ x\ s) \wedge Suc\ y * of\text{-}real\ (ln\ (real\ x))))$
 $abs\text{-}summable\text{-}on\ \{p.\ prime\ p\}$
using *sum2* **by** (*rule abs-summable-on-Sigma-project1'*) *auto*
also have *?this* $\longleftrightarrow (\lambda p. -(of\text{-}real\ (ln\ (real\ p)) * (1 / (1 - fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) - 1)))$ *abs-summable-on* $\{p.\ prime\ p\}$
by (*intro abs-summable-on-cong eq*) *auto*
also have $\dots \longleftrightarrow$ *?th1* **by** (*subst abs-summable-on-uminus-iff*) *auto*
finally show *?th1* .

have *eval-fds* $h\ s = (\sum_a n. fds\text{-}nth\ h\ n / nat\text{-}power\ n\ s)$
using * **unfolding** *eval-fds-def* **by** (*subst infsetsum-nat'*) *auto*
also have $\dots = (\sum_a n \in \{n.\ primepow\ n\}. -fds\text{-}nth\ f\ n * mangoldt\ n / nat\text{-}power\ n\ s)$
unfolding *h-def* **using** *fds-nth-logderiv-completely-multiplicative*[*OF* *assms*(2)]
by (*intro infsetsum-cong-neutral*) (*auto simp: fds-nth-fds mangoldt-def*)
also have $\dots = (\sum_a (p,k) \in (?P \times UNIV). -fds\text{-}nth\ f\ (p \wedge Suc\ k) * mangoldt\ (p \wedge Suc\ k) /$
 $nat\text{-}power\ (p \wedge Suc\ k)\ s)$
using *bij-betw-primepows* **unfolding** *case-prod-unfold*
by (*intro infsetsum-reindex-bij-betw* [*symmetric*])
also have $\dots = (\sum_a (p,k) \in (?P \times UNIV). -((fds\text{-}nth\ f\ p / nat\text{-}power\ p\ s) \wedge Suc\ k) * of\text{-}real\ (ln\ (real\ p)))$
by (*intro infsetsum-cong*)
(auto simp: f.mult f.power mangoldt-def aprimedivisor-prime-power ln-realpow
prime-gt-0-nat
 $nat\text{-}power\text{-}power\text{-}left\ divide\text{-}simps\ simp\ del: power\text{-}Suc)$
also have $\dots = (\sum_a p \mid prime\ p. \sum_a k.$

$$- ((\text{fds-nth } f \ p \ / \ \text{nat-power } p \ s) \wedge \text{Suc } k) * \text{of-real } (\ln (\text{real } p)))$$
using *sum2* **by** (*subst infsetsum-Times*) (*auto simp: case-prod-unfold*)
also have $\dots = (\sum_{a p \mid \text{prime } p} -(\text{of-real } (\ln (\text{real } p)) * (1 / (1 - \text{fds-nth } f \ p \ / \ \text{nat-power } p \ s) - 1))))$
using *eq* **by** (*intro infsetsum-cong*) *auto*
finally show *?th2* **by** (*subst (asm) infsetsum-uminus*)
qed

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** *?th1*)

and $\text{deriv } \text{zeta } s / \text{zeta } s =$
 $-(\sum_{a p \mid \text{prime } p} \text{of-real } (\ln (\text{real } p)) / (p \ \text{powr } s - 1))$ (**is** *?th2*)

proof –

have $*$: *completely-multiplicative-function* (*fds-nth fds-zeta* :: $- \Rightarrow \text{complex}$)
by *standard auto*

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

have $(\lambda p. \ln (\text{real } p) * (1 / (1 - \text{fds-nth } \text{fds-zeta } p \ / \ p \ \text{powr } s) - 1))$
abs-summable-on $\{p. \text{prime } p\}$

using *eval-fds-logderiv-completely-multiplicative*[*OF **, *of s*] *assms* **by** *auto*

also have *?this* $\longleftrightarrow (\lambda p. \ln (\text{real } p) / (p \ \text{powr } s - 1))$ *abs-summable-on* $\{p. \text{prime } p\}$ **using** *assms*

by (*intro abs-summable-on-cong*) (*auto simp: fds-nth-zeta divide-simps dest: prime-gt-1-nat*)

finally show *?th1* .

from *assms* **have** *ev*: *eventually* $(\lambda z. z \in \{z. \text{Re } z > 1\})$ (*nhds s*)

by (*intro eventually-nhds-in-open open-halfspace-Re-gt*) *auto*

have $\text{deriv } \text{zeta } s = \text{deriv } (\text{eval-fds } \text{fds-zeta}) \ s$

by (*intro deriv-cong-ev*[*OF eventually-mono*[*OF ev*]]) (*auto simp: eval-fds-zeta*)

also have $\text{deriv } (\text{eval-fds } \text{fds-zeta}) \ s / \text{zeta } s = \text{eval-fds } (\text{fds-deriv } \text{fds-zeta} / \text{fds-zeta}) \ s$

using *assms zeta-Re-gt-1-nonzero*[*of s*]

by (*subst eval-fds-log-deriv*) (*auto simp: eval-fds-zeta eval-fds-deriv intro!: abscissa*)

also have $\text{eval-fds } (\text{fds-deriv } \text{fds-zeta} / \text{fds-zeta}) \ s =$

$-(\sum_{a p \mid \text{prime } p} \ln (\text{real } p) * (1 / (1 - \text{fds-nth } \text{fds-zeta } p \ / \ p \ \text{powr } s) - 1))$

(**is** $- = -?S$) **using** *eval-fds-logderiv-completely-multiplicative*[*OF **, *of s*] *assms*

by *auto*

also have $?S = (\sum_{a p \mid \text{prime } p} \ln (\text{real } p) / (p \ \text{powr } s - 1))$ **using** *assms*

by (*intro infsetsum-cong*) (*auto simp: fds-nth-zeta divide-simps dest: prime-gt-1-nat*)

finally show *?th2* .

qed

lemma *sums-logderiv-zeta*:

assumes $\text{Re } s > 1$

shows $(\lambda p. \text{if prime } p \text{ then of-real } (\ln (\text{real } p)) / (\text{of-nat } p \text{ powr } s - 1) \text{ else } 0)$
sums

$-(\text{deriv zeta } s / \text{zeta } s) \text{ (is ?f sums -)}$

proof –

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

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

lemma *abs-conv-abcissa-diff-le*:

abs-conv-abcissa $(f - g :: 'a :: \text{dirichlet-series fds}) \leq$

$\max (\text{abs-conv-abcissa } f) (\text{abs-conv-abcissa } g)$

using *abs-conv-abcissa-add-le*[*of f -g*] **by** *auto*

lemma *abs-conv-abcissa-diff-leI*:

abs-conv-abcissa $(f :: 'a :: \text{dirichlet-series fds}) \leq d \implies \text{abs-conv-abcissa } g \leq d$
 \implies

abs-conv-abcissa $(f - g) \leq d$

using *abs-conv-abcissa-diff-le*[*of f g*] **by** (*auto simp: le-max-iff-disj*)

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

proof *safe*

fix x **assume** $x \geq c$

hence $x = x - c + c$ **by** *simp*

thus $x \in \text{range } (\lambda n. n + c)$ **by** *blast*

qed *auto*

lemma *abs-summable-hurwitz-zeta*:

assumes $\text{Re } s > 1$ $a + \text{real } b > 0$

shows $(\lambda n. 1 / (\text{of-nat } n + a) \text{ powr } s)$ *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 Re s a + b*]

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

hence $(\lambda n. 1 / (\text{of-nat } (n + b) + a) \text{ powr } s)$ *abs-summable-on* *UNIV*

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

also have *?this* $\longleftrightarrow (\lambda n. 1 / (\text{of-nat } n + a) \text{ powr } s)$ *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 $a > 0$ **and** $\text{Re } s > 1$

shows *hurwitz-zeta* $(\text{real } a) s = (\sum_a n. \text{of-nat } (n + a) \text{ powr } -s)$

hurwitz-zeta $(\text{real } a) s = (\sum_{a \in \{a..\}} n. \text{of-nat } n \text{ powr } -s)$

proof –

have *hurwitz-zeta* $(\text{real } a) s = (\sum n. \text{of-nat } (n + a) \text{ powr } -s)$

using *assms* **by** (*subst hurwitz-zeta-conv-suminf*) *auto*

also have $\dots = (\sum_a n. \text{of-nat } (n + a) \text{ powr } -s)$
using *abs-summable-hurwitz-zeta*[*of s a 0*] *assms*
by (*intro infsetsum-nat'* [*symmetric*]) (*auto simp: powr-minus field-simps*)
finally show *hurwitz-zeta* (*real a*) *s* = $(\sum_a n. \text{of-nat } (n + a) \text{ powr } -s)$.
also have $\dots = (\sum_a n \in \text{range } (\lambda n. n + a). \text{of-nat } n \text{ powr } -s)$
by (*rule infsetsum-reindex* [*symmetric*]) *auto*
also have $\text{range } (\lambda n. n + a) = \{a..\}$ **by** (*rule range-add-nat*)
finally show *hurwitz-zeta* (*real a*) *s* = $(\sum_a n \in \{a..\}. \text{of-nat } n \text{ powr } -s)$.
qed

lemma *continuous-on-pre-zeta* [*continuous-intros*]:
assumes *continuous-on* *A* *f* *a* > 0
shows *continuous-on* *A* ($\lambda x. \text{pre-zeta } a (f x)$)
proof –
from *assms* **have** *continuous-on UNIV* (*pre-zeta a*)
by (*intro holomorphic-on-imp-continuous-on*[*OF holomorphic-pre-zeta*]) *auto*
from *continuous-on-compose2*[*OF this assms(1)*] **show** *?thesis* **by** *simp*
qed

lemma *continuous-pre-zeta* [*continuous-intros*]:
assumes *continuous* (*at x within A*) *f* *a* > 0
shows *continuous* (*at x within A*) ($\lambda x. \text{pre-zeta } a (f x)$)
proof –
have *continuous* (*at z*) (*pre-zeta a*) **for** *z*
by (*rule continuous-on-interior*[*of UNIV*]) (*insert assms, auto intro!: continuous-intros*)
from *continuous-within-compose3*[*OF this assms(1)*] **show** *?thesis* .
qed

lemma *pre-zeta-bound*:
assumes $0 < \text{Re } s$ **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* = *EM-remainder 1 ?f 0*
have [*simp*]: $-\text{Re } s - 1 = -1 - \text{Re } s$ **by** (*simp add: algebra-simps*)

have $|\text{frac } x - 1 / 2| \leq 1 / 2$ **for** *x* :: *real* **unfolding** *frac-def*
by *linarith*
hence $|\text{pbernpoly } (\text{Suc } 0) x| \leq 1 / 2$ **for** *x*
by (*simp add: pbernpoly-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 $\langle \text{Re } s > 0 \rangle \langle a > 0 \rangle$ **by** *real-asymp*
ultimately have $*$: $\forall x. x \geq \text{real } 0 \longrightarrow \text{norm } (\text{EM-remainder } 1 \text{ ?f } (\text{int } x)) \leq$
 $(1 / 2) / \text{fact } 1 * (-?g (\text{real } x))$
using $\langle a > 0 \rangle \langle \text{Re } s > 0 \rangle$

by (*intro norm-EM-remainder-le-strong-nat'*[**where** $g' = ?g'$ **and** $Y = \{\}$])
 (*auto intro!*: *continuous-intros derivative-eq-intros*
simp: *field-simps norm-mult norm-powr-real-powr add-eq-0-iff*)
have R : $\text{norm } R \leq \text{norm } s / (2 * \text{Re } s) * a \text{ powr } -\text{Re } s$
unfolding R -def **using** *spec*[$OF *$, of 0] **by** *simp*

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

lemma *pre-zeta-bound'*:
assumes $0 < \text{Re } s$ **and** $a > 0$
shows $\text{norm } (\text{pre-zeta } a s) \leq \text{norm } s / (\text{Re } s * a \text{ powr } \text{Re } s)$
proof –
from *assms* **have** $\text{norm } (\text{pre-zeta } a s) \leq (1 + \text{norm } s / \text{Re } s) / 2 * a \text{ powr } -\text{Re } s$
by (*intro pre-zeta-bound*) *auto*
also have $\dots = (\text{Re } s + \text{norm } s) / 2 / (\text{Re } s * a \text{ powr } \text{Re } s)$
using *assms* **by** (*auto simp*: *field-simps powr-minus*)
also have $\text{Re } s + \text{norm } s \leq \text{norm } s + \text{norm } s$ **by** (*intro add-right-mono complex-Re-le-cmod*)
also have $(\text{norm } s + \text{norm } s) / 2 = \text{norm } s$ **by** *simp*
finally show $\text{norm } (\text{pre-zeta } a s) \leq \text{norm } s / (\text{Re } s * a \text{ powr } \text{Re } s)$
using *assms* **by** (*simp add*: *divide-right-mono*)
qed

lemma *summable-comparison-test-bigo*:
fixes $f :: \text{nat} \Rightarrow \text{real}$
assumes *summable* $(\lambda n. \text{norm } (g n))$ $f \in O(g)$
shows *summable* f
proof –
from $\langle f \in O(g) \rangle$ **obtain** C **where** C : *eventually* $(\lambda x. \text{norm } (f x) \leq C * \text{norm } (g x))$ *at-top*
by (*auto elim*: *landau-o.bigE*)
thus *?thesis*
by (*rule summable-comparison-test-ev*) (*insert assms, auto intro: summable-mult*)
qed

lemma *deriv-zeta-eq*:
assumes $s \neq 1$
shows $\text{deriv } \text{zeta } s = \text{deriv } (\text{pre-zeta } 1) s - 1 / (s - 1)^2$
proof –
from s **have** ev : *eventually* $(\lambda z. z \neq 1)$ (*nhds* s) **by** (*intro t1-space-nhds*)

have [derivative-intros]: (pre-zeta 1 has-field-derivative deriv (pre-zeta 1) s) (at s)
by (intro holomorphic-derivI[of - UNIV] holomorphic-intros) auto
have ((λs. pre-zeta 1 s + 1 / (s - 1)) has-field-derivative (deriv (pre-zeta 1) s - 1 / (s - 1)²)) (at s)
using s **by** (auto intro!: derivative-eq-intros simp: power2-eq-square)
also have ?this \longleftrightarrow (zeta has-field-derivative (deriv (pre-zeta 1) s - 1 / (s - 1)²)) (at s)
by (intro has-field-derivative-cong-ev eventually-mono[OF ev]) (auto simp: zeta-def hurwitz-zeta-def)
finally show ?thesis **by** (rule DERIV-imp-deriv)
qed

lemma zeta-remove-zero:

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

lemma eval-fds-deriv-zeta:

assumes $\text{Re } s > 1$
shows eval-fds (fds-deriv fds-zeta) s = deriv zeta s
proof -
have ev: eventually (λz. z ∈ {z. Re z > 1}) (nhds s)
using assms **by** (intro eventually-nhds-in-open open-halfspace-Re-gt) auto
from assms **have** eval-fds (fds-deriv fds-zeta) s = deriv (eval-fds fds-zeta) s
by (subst eval-fds-deriv) auto
also have ... = deriv zeta s
by (intro deriv-cong-ev eventually-mono[OF ev]) (auto simp: eval-fds-zeta)
finally show ?thesis .
qed

lemma length-sorted-list-of-set [simp]:

finite A \implies length (sorted-list-of-set A) = card A
by (metis length-remdups-card-conv length-sort set-sorted-list-of-set sorted-list-of-set-sort-remdups)

lemma le-nat-iff!: $x \leq \text{nat } y \longleftrightarrow x = 0 \wedge y \leq 0 \vee \text{int } x \leq y$

by auto

lemma sum-upto-plus1:

assumes $x \geq 0$
shows sum-upto f (x + 1) = sum-upto f x + f (Suc (nat [x]))
proof -

have $\text{sum-upto } f (x + 1) = \text{sum } f \{0 <.. \text{Suc } (\text{nat } [x])\}$
using *assms* **by** (*simp add: sum-upto-altdef nat-add-distrib*)
also have $\{0 <.. \text{Suc } (\text{nat } [x])\} = \text{insert } (\text{Suc } (\text{nat } [x])) \{0 <.. \text{nat } [x]\}$
by *auto*
also have $\text{sum } f \dots = \text{sum-upto } f x + f (\text{Suc } (\text{nat } [x]))$
by (*subst sum.insert*) (*auto simp: sum-upto-altdef add-ac*)
finally show *?thesis* .
qed

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

lemma *integral-smallo*:
fixes $f g g' :: \text{real} \Rightarrow \text{real}$
assumes $f \in o(g')$ **and** *filterlim g at-top at-top*
assumes $\bigwedge a' x. a \leq a' \implies a' \leq x \implies f \text{ integrable-on } \{a'..x\}$
assumes *deriv*: $\bigwedge x. x \geq a \implies (g \text{ has-field-derivative } g' x) (at x)$
assumes *cont*: *continuous-on* $\{a..\}$ g'
assumes *nonneg*: $\bigwedge x. x \geq a \implies g' x \geq 0$
shows $(\lambda x. \text{integral } \{a..x\} f) \in o(g)$
proof (*rule landau-o.smallI*)
fix $c :: \text{real}$ **assume** $c: c > 0$
note [*continuous-intros*] = *continuous-on-subset* [*OF cont*]
define c' **where** $c' = c / 2$
from c **have** $c': c' > 0$ **by** (*simp add: c'-def*)
from *landau-o.smallD* [*OF assms(1) this*]
obtain b **where** $b: \bigwedge x. x \geq b \implies \text{norm } (f x) \leq c' * \text{norm } (g' x)$
unfolding *eventually-at-top-linorder* **by** *blast*
define b' **where** $b' = \max a b$
define D **where** $D = \text{norm } (\text{integral } \{a..b'\} f)$

have *filterlim* $(\lambda x. c' * g x)$ *at-top at-top*
using c' **by** (*intro filterlim-tendsto-pos-mult-at-top* [*OF tendsto-const*] *assms*)
hence *eventually* $(\lambda x. c' * g x \geq D - c' * g b')$ *at-top*
by (*auto simp: filterlim-at-top*)
thus *eventually* $(\lambda x. \text{norm } (\text{integral } \{a..x\} f) \leq c * \text{norm } (g x))$ *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 $\text{integral } \{a..x\} f = \text{integral } \{a..b'\} f + \text{integral } \{b'..x\} f$
using *elim b'* **by** (*intro Henstock-Kurzweil-Integration.integral-combine* [*symmetric*]
assms) *auto*
also have $\text{norm } \dots \leq D + \text{norm } (\text{integral } \{b'..x\} f)$
unfolding *D-def* **by** (*rule norm-triangle-ineq*)

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

lemma *integral-bigo*:

fixes $f g g' :: \text{real} \Rightarrow \text{real}$
assumes $f \in O(g')$ **and** $\text{filterlim } g \text{ at-top at-top}$
assumes $\bigwedge a' x. a \leq a' \implies a' \leq x \implies f \text{ integrable-on } \{a'..x\}$
assumes $\text{deriv: } \bigwedge x. x \geq a \implies (g \text{ has-field-derivative } g' x) \text{ (at } x \text{ within } \{a..\})$
assumes $\text{cont: continuous-on } \{a..\} g'$
assumes $\text{nonneg: } \bigwedge x. x \geq a \implies g' x \geq 0$
shows $(\lambda x. \text{integral } \{a..x\} f) \in O(g)$
proof –
note $[\text{continuous-intros}] = \text{continuous-on-subset}[OF \text{ cont}]$
from $\text{landau-o.bigE}[OF \text{ assms}(1)]$
obtain $c b$ **where** $c: c > 0$ **and** $b: \bigwedge x. x \geq b \implies \text{norm } (f x) \leq c * \text{norm } (g' x)$
unfolding *eventually-at-top-linorder* **by** *metis*
define c' **where** $c' = c / 2$
define b' **where** $b' = \max a b$
define D **where** $D = \text{norm } (\text{integral } \{a..b'\} f)$
have $\text{filterlim } (\lambda x. c * g x) \text{ at-top at-top}$
using c **by** $(\text{intro } \text{filterlim-tendsto-pos-mult-at-top}[OF \text{ tendsto-const}] \text{ assms})$
hence *eventually* $(\lambda x. c * g x \geq D - c * g b') \text{ at-top}$
by $(\text{auto } \text{simp: filterlim-at-top})$
hence *eventually* $(\lambda x. \text{norm } (\text{integral } \{a..x\} f) \leq 2 * c * \text{norm } (g x)) \text{ at-top}$
using *eventually-ge-at-top*[of b']
proof *eventually-elim*
case $(\text{elim } x)$
have $b': a \leq b' b \leq b'$ **by** $(\text{auto } \text{simp: } b'\text{-def})$
from *elim* b' **have** *integrable*: $(\lambda x. |g' x|) \text{ integrable-on } \{b'..x\}$
by $(\text{intro } \text{integrable-continuous-real } \text{continuous-intros}) \text{ auto}$

have $\text{integral } \{a..x\} f = \text{integral } \{a..b'\} f + \text{integral } \{b'..x\} f$
using $\text{elim } b'$ **by** (*intro Henstock-Kurzweil-Integration.integral-combine [symmetric]*
assms) *auto*
also have $\text{norm } \dots \leq D + \text{norm } (\text{integral } \{b'..x\} f)$
unfolding $D\text{-def}$ **by** (*rule norm-triangle-ineq*)
also have $\text{norm } (\text{integral } \{b'..x\} f) \leq \text{integral } \{b'..x\} (\lambda x. c * \text{norm } (g' x))$
using b' *elim* *assms* c *integrable* **by** (*intro integral-norm-bound-integral b*
assms) *auto*
also have $\dots = c * \text{integral } \{b'..x\} (\lambda x. |g' x|)$ **by** *simp*
also have $\text{integral } \{b'..x\} (\lambda x. |g' x|) = \text{integral } \{b'..x\} g'$
using *assms* b' **by** (*intro integral-cong*) *auto*
also have (g' *has-integral* ($g x - g b'$)) $\{b'..x\}$ **using** b' *elim*
by (*intro fundamental-theorem-of-calculus*)
(auto simp flip: has-field-derivative-iff-has-vector-derivative
intro!: DERIV-subset[OF deriv])
hence $\text{integral } \{b'..x\} g' = g x - g b'$
by (*simp add: has-integral-iff*)
also have $D + c * (g x - g b') \leq 2 * c * g x$
using *elim* **by** (*simp add: field-simps c'-def*)
also have $\dots \leq 2 * c * \text{norm } (g x)$
using c **by** (*intro mult-left-mono*) *auto*
finally show *?case* **by** *simp*
qed
thus *?thesis* **by** (*rule bigoI*)
qed

lemma *primepows-le-subset*:

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

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

lemma *mangoldt-non-primepow*: $\neg \text{primepow } n \implies \text{mangoldt } n = 0$

by (auto simp: mangoldt-def)

lemma *le-imp-bigo-real*:

assumes $c \geq 0$ eventually $(\lambda x. f x \leq c * (g x :: real))$ F eventually $(\lambda x. 0 \leq f x)$ F

shows $f \in O[F](g)$

proof –

have eventually $(\lambda x. norm (f x) \leq c * norm (g x))$ F

using *assms*(2,3)

proof *eventually-elim*

case (*elim* x)

have $norm (f x) \leq c * g x$ **using** *elim* **by** *simp*

also have $\dots \leq c * norm (g x)$ **by** (*intro mult-left-mono* *assms*) *auto*

finally show *?case* .

qed

thus *?thesis* **by** (*intro bigoI*[*of* - *c*]) *auto*

qed

lemma *ln-minus-ln-floor-bigo*: $(\lambda x. \ln x - \ln (real (nat \lfloor x \rfloor))) \in O(\lambda-. 1)$

proof (*intro le-imp-bigo-real*[*of* 1] *eventually-mono*[*OF* *eventually-ge-at-top*[*of* 1]])

fix $x :: real$ **assume** $x \geq 1$

from x **have** $*$: $x - real (nat \lfloor x \rfloor) \leq 1$ **by** *linarith*

from x **have** $\ln x - \ln (real (nat \lfloor x \rfloor)) \leq (x - real (nat \lfloor x \rfloor)) / real (nat \lfloor x \rfloor)$

by (*intro ln-diff-le*) *auto*

also have $\dots \leq 1 / 1$ **using** $*$ **by** (*intro frac-le*) *auto*

finally show $\ln x - \ln (real (nat \lfloor x \rfloor)) \leq 1 * 1$ **by** *simp*

qed *auto*

lemma *cos-geD*:

assumes $\cos x \geq \cos a$ $0 \leq a \leq \pi$ $-\pi \leq x \leq \pi$

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* $a - x$]

by *auto*

qed

lemma *path-image-part-circlepath-same-Re*:

assumes $0 \leq b$ $b \leq \pi$ $a = -b$ $r \geq 0$

shows $path-image (part-circlepath c r a b) = sphere c r \cap \{s. Re s \geq Re c + r * \cos a\}$

proof *safe*

fix z **assume** $z \in path-image (part-circlepath c r a b)$

with *assms* **obtain** *t* **where** $t: t \in \{a..b\}$ $z = c + \text{of-real } r * \text{cis } t$
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** $\text{Re } z \geq \text{Re } c + r * \cos a$
using *cos-monotone-0-pi-le*[of *t b*] *cos-monotone-minus-pi-0*[of *a t*]
by (*cases t ≥ 0*) (*auto intro!: mult-left-mono*)
next
fix *z* **assume** $z: z \in \text{sphere } c \ r \ \text{Re } z \geq \text{Re } c + r * \cos a$
show $z \in \text{path-image } (\text{part-circlepath } c \ r \ a \ b)$
proof (*cases r = 0*)
case *False*
with *assms* **have** $r: r > 0$ **by** *simp*
with *z* **have** $z\text{-eq}: z = c + r * \text{cis } (\text{Arg } (z - c))$
using *Arg-eq*[of $z - c$] **by** (*auto simp: dist-norm exp-eq-polar norm-minus-commute*)
moreover from $z(2) \ r$ *assms* **have** $\cos b \leq \cos (\text{Arg } (z - c))$
by (*subst (asm) z-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*:

$\text{part-circlepath } c \ r \ (x + a) \ (x + b) = (\lambda z. c + \text{cis } x * (z - c)) \circ \text{part-circlepath } c \ r \ a \ b$

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

by (*subst path-image-part-circlepath-same-Re*) (*simp-all add: assms*)

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

proof *safe*

fix *z* **assume** $z: z \in \text{sphere } c \ r$

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

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

by (*auto simp: dist-norm norm-mult norm-minus-commute*
cis-conv-exp exp-minus field-simps norm-divide)

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

lemma *path-image-semicircle-Re-le:*

assumes $r \geq 0$

shows $\text{path-image } (\text{part-circlepath } c \text{ r } (\pi/2) (3/2*\pi)) =$
 $\text{sphere } c \text{ r} \cap \{s. \text{Re } s \leq \text{Re } c\}$

proof –

let $?f = (\lambda z. c + \text{cis } \pi * (z - c))$

have $*: \text{part-circlepath } c \text{ r } (\pi/2) (3/2*\pi) = \text{part-circlepath } c \text{ r } (\pi + (-\pi/2))$
 $(\pi + \pi/2)$

by *simp*

have $\text{path-image } (\text{part-circlepath } c \text{ r } (\pi/2) (3/2*\pi)) =$
 $?f \text{ ' sphere } c \text{ r} \cap ?f \text{ ' } \{s. \text{Re } c \leq \text{Re } s\}$

unfolding $* \text{ part-circlepath-rotate-left path-image-compose path-image-semicircle-Re-ge}$ [*OF*
assms]

by *auto*

also have $?f \text{ ' sphere } c \text{ r} = \text{sphere } c \text{ r}$

by (*rule sphere-rotate*)

also have $?f \text{ ' } \{s. \text{Re } c \leq \text{Re } s\} = \{s. \text{Re } c \geq \text{Re } s\}$

by (*auto simp: image-iff intro!: exI[of - 2 * c - x for x]*)

finally show *?thesis* .

qed

lemma *path-image-semicircle-Im-ge:*

assumes $r \geq 0$

shows $\text{path-image } (\text{part-circlepath } c \text{ r } 0 \pi) =$
 $\text{sphere } c \text{ r} \cap \{s. \text{Im } s \geq \text{Im } c\}$

proof –

let $?f = (\lambda z. c + \text{cis } (\pi/2) * (z - c))$

have $*: \text{part-circlepath } c \text{ r } 0 \pi = \text{part-circlepath } c \text{ r } (\pi / 2 + (-\pi/2)) (\pi / 2$
 $+ \pi/2)$

by *simp*

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

unfolding $* \text{ part-circlepath-rotate-left path-image-compose path-image-semicircle-Re-ge}$ [*OF*
assms]

by *auto*

also have $?f \text{ ' sphere } c \text{ r} = \text{sphere } c \text{ r}$

by (*rule sphere-rotate*)

also have $?f \text{ ' } \{s. \text{Re } c \leq \text{Re } s\} = \{s. \text{Im } c \leq \text{Im } s\}$

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

finally show *?thesis* .

qed

lemma *path-image-semicircle-Im-le:*

assumes $r \geq 0$

shows $\text{path-image } (\text{part-circlepath } c \text{ r } \pi (2 * \pi)) =$

$sphere\ c\ r \cap \{s.\ Im\ s \leq Im\ c\}$

proof –

let $?f = (\lambda z.\ c + cis\ (3 * pi / 2) * (z - c))$

have $*$: $part-circlepath\ c\ r\ pi\ (2 * pi) = part-circlepath\ c\ r\ (3 * pi / 2 + (-pi / 2))$
 $(3 * pi / 2 + pi / 2)$

by *simp*

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

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

by *auto*

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

by *(rule sphere-rotate)*

also have $cis\ (3 * pi / 2) = -i$

using *cis-mult[of pi pi / 2]* **by** *simp*

hence $?f\ '\{s.\ Re\ c \leq Re\ s\} = \{s.\ Im\ c \geq Im\ s\}$

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

finally show $?thesis$.

qed

lemma *powr-numeral [simp]*: $x \geq 0 \implies (x :: real)\ powr\ numeral\ y = x \wedge numeral\ y$

using *powr-numeral[of x y]* **by** *(cases x = 0) auto*

lemma *eval-fds-logderiv-zeta-real*:

assumes $x > (1 :: real)$

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

and $deriv\ zeta\ (of-real\ x) / zeta\ (of-real\ x) =$
 $-of-real\ (\sum_a p \mid prime\ p.\ ln\ (real\ p) / (p\ powr\ x - 1))$ **(is** $?th2$ **)**

proof –

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

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

also have $?this \longleftrightarrow ?th1$

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

finally show $?th1$.

show $?th2$ **using** *assms*

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

qed

lemma

fixes $a\ b\ c\ d :: real$

assumes $ab: d * a + b \geq 1$ **and** $c: c < -1$ **and** $d: d > 0$

defines $C \equiv - ((ln\ (d * a + b) - 1 / (c + 1)) * (d * a + b)\ powr\ (c + 1) / (d * (c + 1)))$

shows *set-integrable-ln-powr-at-top*:

$(\lambda x.\ (ln\ (d * x + b) * ((d * x + b)\ powr\ c)))\ absolutely-integrable-on\ \{a < ..\}$ **(is** $?th1$ **)**

and *set-lebesgue-integral-ln-powr-at-top:*
 $(\int x \in \{a < ..\}. (\ln (d * x + b) * ((d * x + b) \text{ powr } c)) \partial \text{lborel}) = C$ (**is** *?th2*)

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

proof –

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

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

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

proof –

have $1 \leq d * a + b$ **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 $(F \text{ has-field-derivative } f x) (at x) \text{ isCont } f x$ **using** *ab c d gt-1*

by *(auto simp: F-def f-def divide-simps intro!: derivative-eq-intros continuous-intros)*

$(auto \text{ simp: algebra-simps powr-add})?$

show $f x \geq 0$ **using** *gt-1* **by** *(auto simp: f-def)*

qed

have *limits*: $((F \circ \text{real-of-ereal}) \longrightarrow F a) (at\text{-right } (\text{ereal } a))$
 $((F \circ \text{real-of-ereal}) \longrightarrow 0) (at\text{-left } \infty)$

using *c ab d unfolding ereal-tendsto-simps1 F-def* **by** *(real-asymp; simp add: field-simps)+*

have *1*: *set-integrable lborel (einterval a ∞) f* **using** *ab c limits*

by *(intro interval-integral-FTC-nonneg) (auto intro!: * AE-I2)*

thus *2*: *f absolutely-integrable-on {a < ..}*

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

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

by *(intro interval-integral-FTC-nonneg) (auto intro!: *)*

thus *3*: *?th2*

by *(simp add: interval-integral-to-infinity-eq F-def f-def C-def)*

show *?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*: $\ln (\text{fact } n) = \text{sum-upto } \ln n$

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

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

by *(simp add: ln-fact-conv-sum-upto sum-upto-altdef)*

lemma *real-of-nat-div*: $\text{real } (a \text{ div } b) = \text{real-of-int } \lfloor \text{real } a / \text{real } b \rfloor$

by *(subst floor-divide-of-nat-eq) auto*

```

lemma integral-subset-negligible:
  fixes  $f :: 'a :: euclidean-space \Rightarrow 'b :: banach$ 
  assumes  $S \subseteq T$  negligible  $(T - S)$ 
  shows  $\text{integral } S f = \text{integral } T f$ 
proof -
  have  $\text{integral } T f = \text{integral } T (\lambda x. \text{if } x \in S \text{ then } f x \text{ else } 0)$ 
    by (rule integral-spike[of  $T - S$ ]) (use assms in auto)
  also have  $\dots = \text{integral } (S \cap T) f$ 
    by (subst integral-restrict-Int) auto
  also have  $S \cap T = S$  using assms by auto
  finally show ?thesis ..
qed

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

lemma measurable-sum-upto [measurable]:
  fixes  $f :: 'a \Rightarrow nat \Rightarrow real$ 
  assumes [measurable]:  $\bigwedge y. (\lambda t. f t y) \in M \rightarrow_M \text{borel}$ 
  assumes [measurable]:  $x \in M \rightarrow_M \text{borel}$ 
  shows  $(\lambda t. \text{sum-upto } (f t) (x t)) \in M \rightarrow_M \text{borel}$ 
proof -
  have meas:  $(\lambda t. \text{set-lebesgue-integral } \text{l borel } \{y. y \geq 0 \wedge y - \text{real } (\text{nat } \lfloor x t \rfloor) \leq 0\} (\lambda y. f t (\text{nat } \lceil y \rceil)))$ 
     $\in M \rightarrow_M \text{borel}$  (is ?f  $\in -$ ) unfolding set-lebesgue-integral-def
    by measurable
  also have ?f =  $(\lambda t. \text{sum-upto } (f t) (x t))$ 
proof
  fix  $t :: 'a$ 
  show ?f  $t = \text{sum-upto } (f t) (x t)$ 
proof (cases  $x t < 1$ )
  case True
  hence  $\{y. y \geq 0 \wedge y - \text{real } (\text{nat } \lfloor x t \rfloor) \leq 0\} = \{0\}$  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 *:  $((\lambda x. f t (\text{nat } \lceil x \rceil)) \text{ has-integral sum } (f t) \{0 <.. n\}) \{ \text{real } 0 .. \text{real } n \}$ 
    using  $\langle n > 0 \rangle$  by (intro nat-sum-has-integral-ceiling) auto

  have **:  $(\lambda x. f t (\text{nat } \lceil x \rceil))$  absolutely-integrable-on  $\{ \text{real } 0 .. \text{real } n \}$ 
proof (rule absolutely-integrable-absolutely-integrable-ubound)
  show  $(\lambda -. \text{MAX } n \in \{0..n\}. |f t n|)$  absolutely-integrable-on  $\{ \text{real } 0 .. \text{real } n \}$ 

```

```

    using ⟨n > 0⟩ by (subst absolutely-integrable-on-iff-nonneg)
      (auto simp: Max-ge-iff intro!: exI[of - f t 0])
  show (λx. f t (nat ⌈x⌉)) integrable-on {real 0..real n}
    using * by (simp add: has-integral-iff)
next
fix y :: real assume y: y ∈ {real 0..real n}
have f t (nat ⌈y⌉) ≤ |f t (nat ⌈y⌉)|
  by simp
also have ... ≤ (MAX n∈{0..n}. |f t n|)
  using y by (intro Max.coboundedI) auto
finally show f t (nat ⌈y⌉) ≤ (MAX n∈{0..n}. |f t n|) .
qed
have sum (f t) {0<..n} = (∫ x∈{real 0..real n}. f t (nat ⌈x⌉) ∂lebesgue)
  using has-integral-set-lebesgue[OF **] * by (simp add: has-integral-iff)
also have ... = (∫ x∈{real 0..real n}. f t (nat ⌈x⌉) ∂lborel)
  unfolding set-lebesgue-integral-def by (subst integral-completion) auto
also have {real 0..real n} = {y. 0 ≤ y ∧ y - real (nat ⌊x t⌋) ≤ 0}
  by (auto simp: n-def)
also have sum (f t) {0<..n} = sum-upto (f t) (x t)
  by (simp add: sum-upto-altdef n-def)
finally show ?thesis ..
qed
qed
finally show ?thesis .
qed
end

```

2 Ingham's Tauberian Theorem

```

theory Newman-Ingham-Tauberian
imports
  HOL-Real-Asymp.Real-Asymp
  Prime-Number-Theorem-Library
begin

```

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

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

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

```

lemma newman-ingham-aux1:
  fixes R :: real and z :: complex

```

assumes $R: R > 0$ **and** $z: \text{norm } z = R$
shows $\text{norm } (1 / z + z / R^2) = 2 * |\text{Re } z| / R^2$
proof –
from z **and** R **have** $[simp]: z \neq 0$ **by** *auto*
have $1 / z + z / R^2 = (R^2 + z^2) * (1 / R^2 / z)$ **using** R
by $(simp \text{ add: field-simps power2-eq-square})$
also have $\text{norm } \dots = \text{norm } (R^2 + z^2) / R^3$
by $(simp \text{ add: numeral-3-eq-3 } z \text{ norm-divide norm-mult power2-eq-square})$
also have $R^2 + z^2 = z * (z + \text{cnj } z)$ **using** $\text{complex-norm-square}[of \ z]$
by $(simp \text{ add: } z \text{ power2-eq-square algebra-simps})$
also have $\text{norm } \dots = 2 * |\text{Re } z| * R$
by $(subst \text{ complex-add-cnj})$ $(simp\text{-all add: } z \text{ norm-mult})$
also have $\dots / R^3 = 2 * |\text{Re } z| / R^2$
using R **by** $(simp \text{ add: field-simps numeral-3-eq-3 power2-eq-square})$
finally show *?thesis* .
qed

lemma *newman-ingham-aux2*:

fixes $m :: \text{nat}$ **and** $w \ z :: \text{complex}$
assumes $1 \leq m$ $1 \leq \text{Re } w$ $0 < \text{Re } z$ **and** $f: \bigwedge n. 1 \leq n \implies \text{norm } (f \ n) \leq C$
shows $\text{norm } (\sum_{n=1..m}. f \ n / n \ \text{powr } (w - z)) \leq C * (m \ \text{powr } \text{Re } z) * (1 / m + 1 / \text{Re } z)$
proof –
have $[simp]: C \geq 0$ **by** $(rule \ \text{order.trans}[OF - f[of 1]])$ *auto*
have $\text{norm } (\sum_{n=1..m}. f \ n / n \ \text{powr } (w - z)) \leq (\sum_{n=1..m}. C / n \ \text{powr } (1 - \text{Re } z))$
by $(rule \ \text{sum-norm-le})$
 $(insert \ \text{assms, auto simp: norm-divide norm-powr-real-powr intro!: frac-le assms powr-mono})$
also have $\dots = C * (\sum_{n=1..m}. n \ \text{powr } (\text{Re } z - 1))$
by $(subst \ \text{sum-distrib-left})$ $(simp\text{-all add: powr-diff})$
also have $\dots \leq C * (m \ \text{powr } \text{Re } z * (1 / \text{Re } z + 1 / m))$
using $\text{zeta-partial-sum-le}'[of \ \text{Re } z \ m]$ *assms* **by** $(intro \ \text{mult-left-mono})$ *auto*
finally show *?thesis* **by** $(simp \ \text{add: mult-ac add-ac})$
qed

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

fixes $a \ x :: \text{real}$
assumes $ax: a > 0$ $x > 1$
shows $(\sum i. (a + \text{real } (\text{Suc } i)) \ \text{powr } (-x)) \leq a \ \text{powr } (1 - x) / (x - 1)$
proof $(rule \ \text{decreasing-sum-le-integral, goal-cases})$
have $((\lambda t. (a + t) \ \text{powr } -x) \ \text{has-integral } -(a \ \text{powr } (-x + 1)) / (-x + 1))$
 $(interior \ \{0..\})$
using $\text{powr-has-integral-at-top}[of \ 0 \ a \ -x]$ **using** ax **by** $(simp \ \text{add: interior-real-atLeast})$
also have $-(a \ \text{powr } (-x + 1)) / (-x + 1) = a \ \text{powr } (1 - x) / (x - 1)$
using ax **by** $(simp \ \text{add: field-simps})$
finally show $((\lambda t. (a + t) \ \text{powr } -x) \ \text{has-integral } a \ \text{powr } (1 - x) / (x - 1))$
 $\{0..\}$
by $(subst \ (asm) \ \text{has-integral-interior})$ *auto*

qed (*insert ax, auto intro!: pour-mono2*)

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

lemma *analytic-on-axis-extend*:

fixes $y1\ y2\ x :: \text{real}$

defines $S \equiv \{z. \text{Re } z = x \wedge \text{Im } z \in \{y1..y2\}\}$

assumes $y1 \leq y2$

assumes f analytic-on S

obtains $x1\ x2 :: \text{real}$ **where** $x1 < x\ x2 > x$ f analytic-on $\text{cbox } (\text{Complex } x1\ y1)$
($\text{Complex } x2\ y2$)

proof –

define C **where** $C = \{\text{box } a\ b \mid a\ b\ z. f \text{ analytic-on } \text{box } a\ b \wedge z \in \text{box } a\ b \wedge z \in S\}$

have $S = \text{cbox } (\text{Complex } x\ y1)\ (\text{Complex } x\ y2)$

by (*auto simp: S-def in-cbox-complex-iff*)

also have *compact ... by simp*

finally have $1: \text{compact } S$.

have $2: S \subseteq \bigcup C$

proof (*intro subsetI*)

fix z **assume** $z \in S$

from $\langle f \text{ analytic-on } S \rangle$ **and this obtain** $a\ b$ **where** $z \in \text{box } a\ b$ f analytic-on $\text{box } a\ b$

by (*blast elim: analytic-onE-box*)

with $\langle z \in S \rangle$ **show** $z \in \bigcup C$ **unfolding** $C\text{-def}$ **by** *blast*

qed

have $3: \text{open } X$ **if** $X \in C$ **for** X **using that** **by** (*auto simp: C-def*)

from *compactE[OF 1 2 3]* **obtain** T **where** $T: T \subseteq C$ *finite* $T\ S \subseteq \bigcup T$

by *blast*

define $x1$ **where** $x1 = \text{Max } (\text{insert } (x - 1)\ ((\lambda X. x + (\text{Inf } (\text{Re } ' X) - x) / 2)$
' T))

define $x2$ **where** $x2 = \text{Min } (\text{insert } (x + 1)\ ((\lambda X. x + (\text{Sup } (\text{Re } ' X) - x) / 2)$
' T))

have $*$: $x + (\text{Inf } (\text{Re } ' X) - x) / 2 < x \wedge x + (\text{Sup } (\text{Re } ' X) - x) / 2 > x$ **if**
 $X \in T$ **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*)

hence $le: \text{Re } a < \text{Re } b$ $\text{Im } a < \text{Im } b$ **by** (*auto simp: in-box-complex-iff*)

show *?thesis* **using** $le\ s$

unfolding $\langle X = \text{box } a\ b \rangle$ *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** $x1\text{-def}$ **by** (*subst Max-less-iff*) *auto*

from $* T$ **have** $x2 > x$ **unfolding** $x2\text{-def}$ **by** $(subst\ Min\text{-gr}\text{-iff})\ auto$

have f *analytic-on* $(\bigcup T)$
using T **by** $(subst\ analytic\text{-on}\text{-Union})\ (auto\ simp:\ C\text{-def})$

moreover have $z \in \bigcup T$ **if** $z \in cbox\ (Complex\ x1\ y1)\ (Complex\ x2\ y2)$ **for** z
proof –

from that have $Complex\ x\ (Im\ z) \in S$
by $(auto\ simp:\ in\text{-cbox}\text{-complex}\text{-iff}\ S\text{-def})$

with T **obtain** X **where** $X: X \in T\ Complex\ x\ (Im\ z) \in X$
by $auto$

with T **obtain** $a\ b$ **where** $[simp]: X = box\ a\ b$ **by** $(auto\ simp:\ C\text{-def})$

from X **have** $le: Re\ a < Re\ b\ Im\ a < Im\ b$ **by** $(auto\ simp:\ in\text{-box}\text{-complex}\text{-iff})$

from that have $Re\ z \leq x2$ **by** $(simp\ add:\ in\text{-cbox}\text{-complex}\text{-iff})$

also have $\dots \leq x + (Sup\ (Re\ 'X) - x) / 2$
unfolding $x2\text{-def}$ **by** $(rule\ Min.\text{cobounded}I)(use\ T\ X\ in\ auto)$

also have $\dots = (x + Re\ b) / 2$
using le **unfolding** $\langle X = box\ a\ b \rangle\ Re\text{-image}\text{-box}[OF\ le]$ **by** $(simp\ add:\ field\text{-simps})$

also have $\dots < (Re\ b + Re\ b) / 2$
using X **by** $(intro\ divide\text{-strict}\text{-right}\text{-mono}\ add\text{-strict}\text{-right}\text{-mono})$
 $(auto\ simp:\ in\text{-box}\text{-complex}\text{-iff})$

also have $\dots = Re\ b$ **by** $simp$

finally have $[simp]: Re\ z < Re\ b$.

have $Re\ a = (Re\ a + Re\ a) / 2$ **by** $simp$

also have $\dots < (x + Re\ a) / 2$
using X **by** $(intro\ divide\text{-strict}\text{-right}\text{-mono}\ add\text{-strict}\text{-right}\text{-mono})$
 $(auto\ simp:\ in\text{-box}\text{-complex}\text{-iff})$

also have $\dots = x + (Inf\ (Re\ 'X) - x) / 2$
using le **unfolding** $\langle X = box\ a\ b \rangle\ Re\text{-image}\text{-box}[OF\ le]$ **by** $(simp\ add:\ field\text{-simps})$

also have $\dots \leq x1$ **unfolding** $x1\text{-def}$ **by** $(rule\ Max.\text{cobounded}I)(use\ T\ X\ in\ auto)$

also have $\dots \leq Re\ z$ **using** $that$ **by** $(simp\ add:\ in\text{-cbox}\text{-complex}\text{-iff})$

finally have $[simp]: Re\ z > Re\ a$.

from X **have** $z \in X$ **by** $(simp\ add:\ in\text{-box}\text{-complex}\text{-iff})$

with $T\ X$ **show** $?thesis$ **by** $blast$

qed

hence $cbox\ (Complex\ x1\ y1)\ (Complex\ x2\ y2) \subseteq \bigcup T$ **by** $blast$

ultimately have f *analytic-on* $cbox\ (Complex\ x1\ y1)\ (Complex\ x2\ y2)$
by $(rule\ analytic\text{-on}\text{-subset})$

with $\langle x1 < x \rangle$ **and** $\langle x2 > x \rangle$ **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 *coeff-bound*: $\text{fds-nth } F \in O(\lambda \cdot 1)$

assumes *f-analytic*: $f \text{ analytic-on } \{s. \text{Re } s \geq 1\}$

assumes *F-conv-f*: $\bigwedge s. \text{Re } s > 1 \implies \text{eval-fds } F s = f s$

assumes w : $\text{Re } w \geq 1$

shows *fds-converges* $F w$ **and** *eval-fds* $F w = f w$

proof –

– We get a bound on our coefficients and call it C .

obtain C **where** $C: C \geq 1 \bigwedge n. \text{norm } (\text{fds-nth } F n) \leq C$

using *natfun-bigo-1E[OF coeff-bound, where lb = 1]* **by** *blast*

write *contour-integral* ($\oint [-]$)

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

have $(\lambda N. \text{eval-fds } (\text{fds-truncate } N F) w) \longrightarrow f w$

unfolding *tendsto-iff dist-norm norm-minus-commute[of eval-fds F s for F s]*

proof *safe*

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** $R = \max 1 (3 * C / \varepsilon)$

have $R: R \geq 3 * C / \varepsilon \ R \geq 1$ **by** (*auto simp: R-def*)

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

obtain l **where** $l: l > 0$

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

proof –

have *f-analytic'*: $(\lambda z. f(w + z)) \text{ analytic-on } \{s. \text{Re } s \geq 0\}$

by (*rule analytic-on-compose-gen[OF f-analytic, unfolded o-def]*)

(*insert w, auto intro: analytic-intros*)

hence $(\lambda z. f(w + z)) \text{ analytic-on } \{s. \text{Re } s = 0 \wedge \text{Im } s \in \{-R-1 .. R+1\}\}$

```

    by (rule analytic-on-subset) auto
  from analytic-on-axis-extend[OF - this] obtain x1 x2 where x12:
    x1 < 0 x2 > 0 (λz. f (w + z)) analytic-on cbox (Complex x1 (-R-1))
  (Complex x2 (R+1))
  using ⟨R ≥ 1⟩ by auto
  from this(3) have (λz. f (w + z)) analytic-on {s. Re s ∈ {x1..0} ∧ Im s ∈
  {-R-1..R+1}}
  by (rule analytic-on-subset) (insert x12, auto simp: in-cbox-complex-iff)
  with f-analytic' have (λz. f (w + z)) analytic-on
    ({s. Re s ≥ 0} ∪ {s. Re s ∈ {x1..0} ∧ Im s ∈
  {-R-1..R+1}})
  by (subst analytic-on-Un) auto
  hence (λz. f (w + z)) analytic-on {s. Re s > 0 ∨ Im s ∈ {-R-1<..

```

— 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 l have [simp]: 0 ∈ X by (auto simp: X-def in-box-complex-iff)
have analytic: (λz. 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, un-
folded 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 ana-
lytic]]]

```

— 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 δ where δ = min (1/2) (l/2)
from l have δ: δ > 0 δ ≤ 1/2 δ < l by (auto simp: δ-def)
define X' where X' = cbox (Complex (-δ) (-R)) (Complex R R)
have X' ⊆ X unfolding X'-def X-def using l(1) R δ
  by (intro subset-box-imp) (auto simp: Basis-complex-def)
have [intro]: compact X' by (simp add: X'-def)
moreover have continuous-on X' (λz. f (w + z))
  using w ⟨X' ⊆ X⟩ by (auto intro!: continuous-intros)
ultimately obtain M where M: M ≥ 0 ∧ z. z ∈ X' ⇒ norm (f (w + z))
≤ M
  using continuous-on-compact-bound by blast

```

— Our objective is now to show that the difference between the N -th partial sum and the limit is below a certain bound (depending on N) which tends to 0 for $N \rightarrow \infty$. We use the following bound:

```

define bound where
  bound = ( $\lambda N::nat.$  ( $2 * C / R + C / N + 3 * M / (pi * R * ln N) + 3 * R * M / (\delta * pi * N$ 
   $powr \delta)$ ))
have  $2 * C / R < \varepsilon$  using  $M(1) R C(1) \delta(1) \varepsilon$ 
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  $M(1) R C(1) \delta(1) \varepsilon$  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 ( $f w - eval-fds (fds-truncate N F) w$ )  $< \varepsilon$ ) 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 Γ consists of a semicircle A of radius R in the right-halfplane and a box of width δ 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  $B1 = linepath (R * i) (R * i - \delta)$ 
define B3 where  $B3 = linepath (-R * i - \delta) (-R * i)$ 
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 =  $-i * R$  pathfinish A =  $i * R$ 
  pathstart B1 =  $i * R$  pathfinish B1 =  $R * i - \delta$ 
  pathstart B3 =  $-R * i - \delta$  pathfinish B3 =  $-i * R$ 
  pathstart B2 =  $R * i - \delta$  pathfinish B2 =  $-R * i - \delta$  using  $R \delta$ 
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 exp-eq-polar 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* $B1 = \{s. \text{Re } s \in \{-\delta..0\} \wedge \text{Im } s = R\}$
and *image-B3*: *path-image* $B3 = \{s. \text{Re } s \in \{-\delta..0\} \wedge \text{Im } s = -R\}$
using δ **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-semicircle-Re-ge*) *auto*
also have $z \in \dots \rightarrow z \in X' - \{0\}$ **for** z
using *complex-Re-le-cmod*[*of* z] *abs-Im-le-cmod*[*of* z] δ 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* $B2 \subseteq X' - \{0\}$ *path-image* $A \subseteq X' - \{0\}$
path-image $B1 \subseteq X' - \{0\}$ *path-image* $B3 \subseteq X' - \{0\}$ **using** $\langle \delta > 0 \rangle$
by (*auto simp: X'-def in-cbox-complex-iff image-B2 image-B1 image-B3*)
note *path-images = this* $\langle X' \subseteq X \rangle$

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

from R **have** *simple-path* A **unfolding** *A-def*
by (*subst simple-path-part-circlepath*) *auto*
have *simple-path* Γ **unfolding** *\Gamma-def*
proof (*intro simple-path-join-loop subsetI arc-join, goal-cases*)
fix z **assume** $z: z \in \text{path-image } A \cap \text{path-image } (B1 \text{ +++ } B2 \text{ +++ } B3)$
with *image-A* **have** $\text{Re } z \geq 0$ $\text{norm } z = R$ **by** *auto*
with z R δ **show** $z \in \{\text{pathstart } A, \text{pathstart } (B1 \text{ +++ } B2 \text{ +++ } B3)\}$
by (*auto simp: path-image-join image-B1 image-B2 image-B3 complex-eq-iff*)
qed (*insert R, auto simp: image-B1 image-B3 path-image-join image-B2 complex-eq-iff*)

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

define g **where** $g = (\lambda z::\text{complex}. f (w + z) * N \text{ powr } z * (1 / z + z / R^2))$
define S **where** $S = \text{eval-fds } (\text{fds-truncate } N F)$
define $g\text{-}S$ **where** $g\text{-}S = (\lambda z::\text{complex}. S (w + z) * N \text{ powr } z * (1 / z + z / R^2))$
define rem **where** $\text{rem} = (\lambda z::\text{complex}. f z - S z)$
define $g\text{-rem}$ **where** $g\text{-rem} = (\lambda z::\text{complex}. \text{rem } (w + z) * N \text{ powr } z * (1 / z + z / R^2))$

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*: *abs-conv-abscissa* $F \leq 1$
using *assms* **by** (*intro bounded-coeffs-imp-abs-conv-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 $eval-fds \dots z = S z + eval-fds (fds-remainder N F) z$
unfolding $S-def$
by ($subst\ eval-fds-add$) ($auto\ intro!$: $fds-abs-converges-imp-converges$
 $fds-abs-converges[OF\ le-less-trans[OF\ abscissa]]$)
finally show $?thesis$ **by** ($simp\ add: rem-def$)
qed

— We now come to the first application of the residue theorem along the path

Γ :

have $\oint[\Gamma] g = 2 * pi * i * winding-number\ \Gamma\ 0 * residue\ g\ 0$
proof ($subst\ Residue-theorem$)
show g *holomorphic-on* $X - \{0\}$ **by** *fact*
show $path-image\ \Gamma \subseteq X - \{0\}$ **using** $path-images$
by ($auto\ simp: \Gamma-def\ path-image-join$)
thus $\forall z. z \notin X \longrightarrow winding-number\ \Gamma\ z = 0$
by ($auto\ intro!$: $simply-connected-imp-winding-number-zero[of\ X]$
 $convex-imp-simply-connected$)
qed ($insert\ path-images, auto\ intro: convex-connected$)
also have $winding-number\ \Gamma\ 0 = 1$
proof ($rule\ simple-closed-path-winding-number-pos$)
from $R\ \delta$ **have** $\forall g \in \{A, B1, B2, B3\}. Re (winding-number\ g\ 0) > 0$
unfolding $A-def\ B1-def\ B2-def\ B3-def$
by ($auto\ intro!$: $winding-number-linepath-pos-lt\ winding-number-part-circlepath-pos-less$)
hence $valid-path\ \Gamma \wedge 0 \notin path-image\ \Gamma \wedge Re (winding-number\ \Gamma\ 0) > 0$
unfolding $\Gamma-def$ **using** $path-images(1-4)$ **by** ($intro\ winding-number-join-pos-combined'$)

auto

thus $Re (winding-number\ \Gamma\ 0) > 0$ **by** *simp*
qed ($insert\ path-images\ \langle simple-path\ \Gamma \rangle, auto\ simp: \Gamma-def\ path-image-join$)
also have $residue\ g\ 0 = f\ w$
proof —
have $g = (\lambda z::complex. f (w + z) * N\ powr\ z * (1 + z^2 / R^2) / z)$
by ($auto\ simp: g-def\ divide-simps\ fun-eq-iff\ power2-eq-square$
 $simp\ del: div-mult-self3\ div-mult-self4\ div-mult-self2\ div-mult-self1$)
moreover from N **have** $residue \dots 0 = f\ w$
by ($subst\ residue-simple'[of\ X]$)
 $(auto\ intro!: holomorphic-intros\ analytic-imp-holomorphic[OF\ analytic])$
ultimately show $?thesis$ **by** (*simp only*:)
qed
finally have $2 * pi * i * f\ w = \oint[\Gamma] g$ **by** *simp*
also have $\dots = \oint[A] g + \oint[B2] g + \oint[B1] g + \oint[B3] g$ **unfolding** $\Gamma-def$
by ($subst\ contour-integral-join, (insert\ path-images,$
 $auto\ intro!: contour-integral-join\ contour-integrable-holomorphic-simple$
 $g-holo)[4])+$
 $(simp-all\ add: add-ac)$)
finally have $integral1: 2 * pi * i * f\ w = \oint[A] g + \oint[B2] g + \oint[B1] g +$
 $\oint[B3] g .$

— Next, we apply the residue theorem along a circle of radius R to another

integrand that is related to the partial sum:

```

have  $\oint$  [circlepath 0 R] g-S = 2 * pi * i * residue g-S 0
proof (subst Residue-theorem)
  show g-S holomorphic-on UNIV - {0}
  by (auto simp: g-S-def S-def intro!: holomorphic-intros)
qed (insert R, auto simp: winding-number-circlepath-centre)
also have residue g-S 0 = S w
proof -
  have g-S = ( $\lambda z::\text{complex. } S (w + z) * N \text{ pow } z * (1 + z^2 / R^2) / z$ )
  by (auto simp: g-S-def divide-simps fun-eq-iff power2-eq-square
    simp del: div-mult-self3 div-mult-self4 div-mult-self2 div-mult-self1)
  moreover from N have residue ... 0 = S w
  by (subst residue-simple'[of X])
    (auto intro!: holomorphic-intros simp: S-def)
  ultimately show ?thesis by (simp only)
qed
finally have 2 * pi * i * S w =  $\oint$  [circlepath 0 R] g-S ..

```

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

```

also have ... =  $\oint$  [part-circlepath 0 R (-pi/2) (3*pi/2)] g-S
proof (rule Cauchy-theorem-homotopic-loops)
  show homotopic-loops (-{0}) (circlepath 0 R)
    (part-circlepath 0 R (- pi / 2) (3 * pi / 2)) unfolding circlepath-def
using R
  by (intro homotopic-loops-part-circlepath[where k = 1]) auto
qed (auto simp: g-S-def S-def intro!: holomorphic-intros)
also have ... =  $\oint$  [A +++ -A] g-S
proof (rule Cauchy-theorem-homotopic-paths)
  have *: -A = part-circlepath 0 R (pi/2) (3*pi/2) unfolding A-def
  by (intro part-circlepath-mirror[where k = 0]) auto
  from R show homotopic-paths (-{0}) (part-circlepath 0 R (-pi/2)
    (3*pi/2)) (A +++ -A)
  unfolding * unfolding A-def
  by (intro homotopic-paths-part-circlepath) (auto dest!: in-path-image-part-circlepath)
qed (auto simp: g-S-def S-def A-def exp-eq-polar intro!: holomorphic-intros)
also have ... =  $\oint$  [A] g-S +  $\oint$  [-A] g-S using R
  by (intro contour-integral-join contour-integrable-holomorphic-simple[of -
    -{0}])
    (auto simp: A-def g-S-def S-def path-image-mirror dest!: in-path-image-part-circlepath
      intro!: holomorphic-intros)
also have  $\oint$  [-A] g-S = - $\oint$  [A] ( $\lambda x. g-S (-x)$ )
  by (simp add: A-def contour-integral-mirror contour-integral-neg)
finally have integral2: 2 * pi * i * S w =  $\oint$  [A] g-S -  $\oint$  [A] ( $\lambda x. g-S (-x)$ )
  by simp

```

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

```

have circle-bound: norm (1 / z + z / R2) ≤ 2 / R if [simp]: norm z = R

```



```

for  $z :: \text{complex}$ 
  proof –
    have  $\text{norm } (1 / z + z / R^2) \leq 1 / R + 1 / R$ 
      by (intro order.trans[OF norm-triangle-ineq] add-mono)
        (insert R, simp-all add: norm-divide norm-mult power2-eq-square)
    thus ?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.
  have  $B12\text{-bound: norm } (\text{integral } \{-\delta..0\} (\lambda x. g (x + R' * i))) \leq 3 * M / R / \ln N$ 
    (is ?I ≤ -) if  $|R'| = R$  for  $R'$ 
  proof –
    have  $?I \leq \text{integral } \{-\delta..0\} (\lambda x. 3 * M / R * N \text{ powr } x)$ 
    proof (rule integral-norm-bound-integral)
      fix  $x$  assume  $x: x \in \{-\delta..0\}$ 
      define  $z$  where  $z = x + i * R'$ 
      from  $R$  that have [simp]:  $z \neq 0$   $\text{Re } z = x$   $\text{Im } z = R'$ 
      by (auto simp: z-def complex-eq-iff)
    from  $x R$  that have  $z \in X'$  by (auto simp: z-def X'-def in-cbox-complex-iff)
    from  $x R$  that have  $\text{norm } z \leq \delta + R$ 
      by (intro order.trans[OF cmod-le add-mono]) auto

    hence  $\text{norm } (1 / z + z / R^2) \leq 1 / R + (\delta / R + 1) / R$ 
      using  $R$  that abs-Im-le-cmod[of z]
      by (intro order.trans[OF norm-triangle-ineq add-mono])
        (auto simp: norm-divide norm-mult power2-eq-square field-simps )
    also have  $\delta / R \leq 1$  using  $\delta R$  by auto
    finally have  $\text{norm } (1 / z + z / R^2) \leq 3 / R$ 
      using  $R$  by (simp add: divide-right-mono)
    hence  $\text{norm } (g z) \leq M * N \text{ powr } x * (3 / R)$ 
      unfolding  $g\text{-def}$  norm-mult using  $\langle M \geq 0 \rangle \langle z \in X' \rangle$ 
    by (intro mult-mono mult-nonneg-nonneg M) (auto simp: norm-powr-real-powr)
    thus  $\text{norm } (g (x + R' * i)) \leq 3 * M / R * N \text{ powr } x$  by (simp add:
mult-ac z-def)
  qed (insert N R l that  $\delta$ , auto intro!: integrable-continuous-real continuous-intros
      simp:  $g\text{-def}$  X-def complex-eq-iff in-box-complex-iff)
  also have  $\dots = 3 * M / R * \text{integral } \{-\delta..0\} (\lambda x. N \text{ powr } x)$  by simp
  also have  $((\lambda x. N \text{ powr } x) \text{ has-integral } (N \text{ powr } 0 / \ln N - N \text{ powr } (-\delta) / \ln N)) \{-\delta..0\}$ 
    using  $\delta N$ 
    by (intro fundamental-theorem-of-calculus)
      (auto simp: has-field-derivative-iff-has-vector-derivative [symmetric]
powr-def
      intro!: derivative-eq-intros)
  hence  $\text{integral } \{-\delta..0\} (\lambda x. N \text{ powr } x) = 1 / \ln (\text{real } N) - \text{real } N \text{ powr } -\delta / \ln (\text{real } N)$ 

```

using N **by** (*simp add: has-integral-iff*)
also have $\dots \leq 1 / \ln (\text{real } N)$ **using** N **by** *simp*
finally show *?thesis* **using** $M R$ **by** (*simp add: mult-left-mono divide-right-mono*)
qed

— We combine the two results from the residue theorem and obtain an integral representation of the difference between the partial sums and the limit:

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

— We now bound each of these integrals individually:

also have $\text{norm } \dots \leq 2 * C * \pi / R + 2 * C * \pi * (1 / N + 1 / R)$
 $+ 3 * M / R / \ln N +$
 $3 * M / R / \ln N + 6 * R * M * N \text{ powr } (-\delta) / \delta$
proof (*rule order.trans[OF norm-triangle-ineq] add-mono*)
have $\oint [B1] g = -\oint [\text{reversepath } B1] g$ **by** (*simp add: contour-integral-reversepath*)
also have $\oint [\text{reversepath } B1] g = \text{integral } \{-\delta..0\} (\lambda x. g (x + R * i))$
unfolding *B1-def reversepath-linepath* **using** δ
by (*subst contour-integral-linepath-same-Im*) *auto*
also have $\text{norm } (-\dots) = \text{norm } \dots$ **by** *simp*
also have $\dots \leq 3 * M / R / \ln N$ **using** R **by** (*intro B12-bound*) *auto*
finally show $\text{norm } (\oint [B1] g) \leq \dots .$
next
have $\oint [B3] g = \text{integral } \{-\delta..0\} (\lambda x. g (x + (-R) * i))$ **unfolding** *B3-def*
using δ
by (*subst contour-integral-linepath-same-Im*) *auto*
also have $\text{norm } \dots \leq 3 * M / R / \ln N$ **using** R **by** (*intro B12-bound*)
auto
finally show $\text{norm } (\oint [B3] g) \leq \dots .$
next
have $\text{norm } (\oint [B2] g) \leq M * N \text{ powr } (-\delta) * (3 / \delta) *$
 $\text{norm } (\text{Complex } (-\delta) (-R) - \text{Complex } (-\delta) R)$ **unfolding** *B2-def*
proof (*(rule contour-integral-bound-linepath; (fold B2-def)?), goal-cases*)
case ($3 z$)
from $3 \delta R$ **have** [*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-ivl*)

```

from  $\mathfrak{B}$  have  $z \in X'$  using  $R \delta$  path-images by (auto simp: B2-def)
from  $\mathfrak{B} \delta R$  have  $\text{norm } z \leq \text{sqrt } (\delta^2 + R^2)$  unfolding cmod-def using
Re-z Im-z
by (intro real-sqrt-le-mono add-mono) (auto simp: power2-le-iff-abs-le)
from power-mono[OF this, of 2] have  $\text{norm } z^2 \leq \delta^2 + R^2$ 
by simp

have  $\text{norm } (1/z + z/R^2) \leq (1 + (\text{norm } z)^2 / R^2) / \delta$ 
unfolding add-divide-distrib using  $\delta R$  abs-Re-le-cmod[of z]
by (intro order.trans[OF norm-triangle-ineq] add-mono)
(auto simp: norm-divide norm-mult field-simps power2-eq-square Re-z)
also have  $\dots \leq (1 + (1 + \delta^2 / R^2)) / \delta$  using  $\delta R \langle z \in X' \rangle$  norm-sqr
unfolding X'-def
by (intro divide-right-mono add-left-mono)
(auto simp: field-simps in-cbox-complex-iff intro!: power-mono)
also have  $\delta^2 / R^2 \leq 1$ 
using  $\delta R$  by (auto simp: field-simps intro!: power-mono)
finally have  $\text{norm } (1/z + z/R^2) \leq \mathfrak{B} / \delta$  using  $\delta$  by (simp add:
divide-right-mono)
with  $\langle z \in X' \rangle$  show  $\text{norm } (g z) \leq M * N \text{ powr } (-\delta) * (\mathfrak{B} / \delta)$  unfolding
g-def norm-mult
by (intro mult-mono mult-nonneg-nonneg M) (auto simp: norm-powr-real-powr
Re-z)
qed (insert path-images M \delta, auto intro!: contour-integrable-holomorphic-simple[OF
g-holo])
thus  $\text{norm } (\oint [B2] g) \leq 6 * R * M * N \text{ powr } (-\delta) / \delta$ 
using  $R$  by (simp add: field-simps cmod-def real-sqrt-mult)
next
have  $\text{norm } (\oint [A] (\lambda x. g-S (-x))) \leq (2 * C / (\text{real } N * R) + 2 * C / R^2)$ 
*

$$R * ((\pi/2) - (-\pi/2))$$
 unfolding A-def
proof ((rule contour-integral-bound-part-circlepath-strong[where k = {R *
i, -R*i}]);
(fold A-def)?, goal-cases)
case ( $6 z$ )
hence [simp]:  $z \neq 0$  and  $\text{norm } z = R$  using  $R$ 
by (auto simp: A-def dest!: in-path-image-part-circlepath)
from  $6$  have  $\text{Re } z \neq 0$ 
using  $\langle \text{norm } z = R \rangle$  by (auto simp: cmod-def abs-iff complex-eq-iff split:
if-splits)
with  $6$  have  $\text{Re } z > 0$  using image-A by auto
have  $S (w - z) = (\sum k = 1..N. \text{fds-nth } F k / \text{of-nat } k \text{ powr } (w - z))$ 
by (simp add: S-def eval-fds-truncate)
also have  $\text{norm } \dots \leq C * N \text{ powr } \text{Re } z * (1 / N + 1 / \text{Re } z)$ 
using  $\langle \text{Re } z > 0 \rangle$   $w N$  by (intro newman-ingham-aux2 C) auto
finally have  $\text{norm } (S (w - z)) \leq \dots$ 
hence  $\text{norm } (g-S (-z)) \leq$ 
 $(C * N \text{ powr } (\text{Re } z) * (1 / N + 1 / \text{Re } z)) * N \text{ powr } (-\text{Re } z) * (2$ 
*  $\text{Re } z / R^2)$ 

```

```

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

also have  $\dots = (\sum n. C * (\text{Suc } n + N) \text{ powr } -\text{Re } (w + z))$ 
unfolding powr-minus by (simp add: field-simps)
also have  $\dots = C * (\sum n. (\text{Suc } n + N) \text{ powr } -\text{Re } (w + z))$ 
using summable-hurwitz-zeta-real[of Re w + Re z Suc N]  $\langle \text{Re } z > 0 \rangle$  w
by (subst suminf-mult) (auto simp: add-ac)
also have  $(\sum n. (\text{Suc } n + N) \text{ powr } -\text{Re } (w + z)) \leq$ 

```

```

      N powr (1 - Re (w + z)) / (Re (w + z) - 1)
    using ⟨Re z > 0⟩ w N hurwitz-zeta-real-bound-aux[of N Re (w + z)]
    by (auto simp: add-ac)
  also have ... ≤ N powr -Re z / Re z
    using w N ⟨Re z > 0⟩ by (intro frac-le powr-mono) auto
  finally have norm (rem (w + z)) ≤ C / (Re z * N powr Re z)
    using C by (simp add: mult-left-mono mult-right-mono powr-minus
field-simps)
  hence norm (g-rem z) ≤ (C / (Re z * N powr Re z)) * N powr (Re z) *
(2 * Re z / R2)
    unfolding g-rem-def norm-mult
    using newman-ingham-aux1[OF - ⟨norm z = R⟩] R ⟨Re z > 0⟩ C
    by (intro mult-mono mult-nonneg-nonneg circle-bound)
      (auto simp: norm-powr-real-powr norm-uminus-minus)
  also have ... = 2 * C / R2 using R N ⟨Re z > 0⟩
    by (simp add: powr-minus field-simps)
  finally show ?case .
next
  show g-rem contour-integrable-on A using path-images
    by (auto simp: g-rem-def rem-def S-def
intro!: contour-integrable-holomorphic-simple[of - X-{0}])
holomorphic-intros)
  qed (insert R N C, auto)
  also have (2 * C / R2) * R * ((pi/2) - (-pi/2)) = 2 * C * pi / R
    using R by (simp add: power2-eq-square field-simps)
  finally show norm (∫ [A] g-rem) ≤ ... .
qed
  also have ... = 4*C*pi/R + 2*C*pi/N + 6*M/R / ln N + 6*R*M*N
powr - δ / δ
    by (simp add: algebra-simps)
  also have ... = 2*pi * (2*C/R + C/N + 3*M / (pi*R*ln N) + 3*R*M
/ (δ*pi * N powr δ))
    by (simp add: field-simps powr-minus)
  also have norm (2 * pi * i * (f w - S w)) = 2 * pi * norm (f w - S w)
    by (simp add: norm-mult)
  finally have norm (f w - S w) ≤ bound N by (simp add: bound-def)
  also have bound N < ε by fact
  finally show norm (f w - S w) < ε .
qed
qed
thus fds-converges F w
  by (auto simp: fds-converges-altdef2 intro: convergentI)
thus eval-fds F w = f w
  using ⟨λN. eval-fds (fds-truncate N F) w ⟶ f w⟩
  by (intro tendsto-unique[OF - tendsto-eval-fds-truncate]) auto
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.

theorem *Newman-Ingham*:

fixes $F :: \text{complex fds}$ **and** $f :: \text{complex} \Rightarrow \text{complex}$

assumes *coeff-bound*: $\text{fds-nth } F \in O(\lambda n. n \text{ powr of-real } (\sigma - 1))$

assumes *f-analytic*: $f \text{ analytic-on } \{s. \text{Re } s \geq \sigma\}$

assumes *F-conv-f*: $\bigwedge s. \text{Re } s > \sigma \implies \text{eval-fds } F s = f s$

assumes *w*: $\text{Re } w \geq \sigma$

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* $F' = (\lambda n. \text{fds-nth } F n * \text{of-nat } n \text{ powr } -\text{of-real}(\sigma - 1))$

by (*auto simp: fun-eq-iff F'-def*)

also have $\dots \in O(\lambda n. \text{of-nat } n \text{ powr of-real } (\sigma - 1) * \text{of-nat } n \text{ powr } -\text{of-real}(\sigma - 1))$

by (*intro landau-o.big.mult-right assms*)

also have $(\lambda n. \text{of-nat } n \text{ powr of-real } (\sigma - 1) * \text{of-nat } n \text{ powr } -\text{of-real } (\sigma - 1)) \in \Theta(\lambda-. 1)$

by (*intro bigthetaI-cong eventually-mono[OF eventually-gt-at-top[of 0]]*)

(*auto simp: powr-minus powr-diff*)

finally have *bigo*: $\text{fds-nth } F' \in O(\lambda-. 1)$.

from *f-analytic* **have** *analytic*: $f' \text{ analytic-on } \{s. \text{Re } s \geq 1\}$ **unfolding** *f'-def*

by (*intro analytic-on-compose-gen[OF f-analytic]*) (*auto intro!: analytic-intros*)

have $F'-f$: *eval-fds* $F' s = f' s$ **if** $\text{Re } s > 1$ **for** s

using *assms that* **by** (*auto simp: F'-def f'-def algebra-simps*)

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

using w **by** *simp*

have 1: *fds-converges* $F' (w - \text{of-real } (\sigma - 1))$

using *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 *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

theory *Prime-Counting-Functions*

imports *Prime-Number-Theorem-Library*
begin

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

3.1 Definitions

definition *prime-sum-upto* :: (nat \Rightarrow 'a) \Rightarrow real \Rightarrow 'a :: *semiring-1* **where**
prime-sum-upto f x = (\sum p | prime p \wedge real p \leq 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 = (\sum p | prime p \wedge p \leq nat \lfloor x \rfloor . f p)

unfolding *sum-upto-altdef prime-sum-upto-altdef1*

by (*intro sum.mono-neutral-cong-right*) (*auto simp: ind-def dest: prime-gt-1-nat*)

lemma *prime-sum-upto-altdef3*:

prime-sum-upto f x = (\sum p \leftarrow primes-upto (nat \lfloor x \rfloor). f p)

proof –

have (\sum p \leftarrow primes-upto (nat \lfloor x \rfloor). f p) = (\sum p | prime p \wedge p \leq nat \lfloor x \rfloor . f p)

by (*subst sum-list-distinct-conv-sum-set*) (*auto simp: set-primes-upto conj-commute*)

thus *?thesis* **by** (*simp add: prime-sum-upto-altdef2*)

qed

lemma *prime-sum-upto-eqI*:

assumes a \leq b \wedge k. k \in {nat \lfloor a \rfloor <.. nat \lfloor b \rfloor } \implies \neg prime k

shows *prime-sum-upto* f a = *prime-sum-upto* f b

proof –

have *: k \leq nat \lfloor a \rfloor **if** k \leq nat \lfloor b \rfloor **prime** k **for** k

using *that assms(2)[of k]* **by** (*cases k \leq nat \lfloor a \rfloor*) *auto*

from *assms(1)* **have** nat \lfloor a \rfloor \leq nat \lfloor b \rfloor **by** *linarith*

hence (\sum p | prime p \wedge p \leq nat \lfloor a \rfloor . f p) = (\sum p | prime p \wedge p \leq nat \lfloor b \rfloor . f p)

using *assms* **by** (*intro sum.mono-neutral-left*) (*auto dest: **)

thus *?thesis* **by** (*simp add: prime-sum-upto-altdef2*)

qed

lemma *prime-sum-upto-eqI'*:

assumes a' \leq nat \lfloor a \rfloor a \leq b nat \lfloor b \rfloor \leq b' \wedge k. k \in {a' <.. b'} \implies \neg 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*
 ($\lambda p. \text{of-nat } (f\ p)$) *x*
by (*simp add: prime-sum-upto-def*)

lemma *prime-sum-upto-mono*:
assumes $\bigwedge n. n > 0 \implies f\ n \geq (0 :: \text{real})$ $x \leq y$
shows *prime-sum-upto* *f* *x* \leq *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 $\bigwedge n. n > 0 \implies f\ n \geq (0 :: \text{real})$
shows *prime-sum-upto* *f* *x* ≥ 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 *prime-sum-upto* *f* *x* = 0
proof –
from *assms* **have** $\text{nat } \lfloor x \rfloor = 0 \vee \text{nat } \lfloor x \rfloor = 1$ **by** *linarith*
thus *?thesis* **by** (*auto simp: eval-prime-sum-upto*)
qed

lemma *measurable-prime-sum-upto* [*measurable*]:
fixes $f :: 'a \Rightarrow \text{nat} \Rightarrow \text{real}$
assumes [*measurable*]: $\bigwedge y. (\lambda t. f\ t\ y) \in M \rightarrow_M \text{borel}$
assumes [*measurable*]: $x \in M \rightarrow_M \text{borel}$
shows $(\lambda t. \text{prime-sum-upto } (f\ t) (x\ t)) \in M \rightarrow_M \text{borel}$
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 $\bigwedge n. \neg \text{primepow } n \implies f\ n = 0 \wedge \bigwedge p\ i. \text{prime } p \implies i > 0 \implies f\ (p \wedge i) = g\ p\ i$
shows *sum-upto* *f* *x* = $(\sum (p, i) \mid \text{prime } p \wedge i > 0 \wedge \text{real } (p \wedge i) \leq x. g\ p\ i)$
proof –
let *?d* = *aprimedivisor*
have $g: g\ (?d\ n) (\text{multiplicity } (?d\ n)\ n) = f\ n$ **if** *primepow* *n* **for** *n* **using** *that*
by (*subst assms(2) [symmetric]*)
(auto simp: primepow-decompose aprimedivisor-prime-power primepow-gt-Suc-0 intro!: aprimedivisor-nat multiplicity-aprimedivisor-gt-0-nat)
have *sum-upto* *f* *x* = $(\sum n \mid \text{primepow } n \wedge \text{real } n \leq x. f\ n)$
unfolding *sum-upto-def* **using** *assms*


```

    by (intro sum.mono-neutral-cong-right) (auto simp: primepow-gt-0-nat)
  also have ... = (∑ (p, i) | prime p ∧ i > 0 ∧ real (p ^ i) ≤ x. g p i) (is - =
sum - ?S)
  by (rule sum.reindex-bij-witness[of - λ(p,i). p ^ i λn. (?d n, multiplicity (?d n)
n)])
  (auto simp: aprime divisor-prime-power primepow-decompose primepow-gt-Suc-0
g
simp del: of-nat-power intro!: aprime divisor-nat multiplicity-aprime divisor-gt-0-nat)
  finally show ?thesis .
qed

```

```

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

```

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

```

bundle prime-counting-notation
begin

```

```

notation primes-pi   (π)
notation primes-theta (∅)
notation primes-psi   (ψ)
notation primes-M     (ℳ)

```

```

end

```

```

bundle no-prime-counting-notation
begin

```

```

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

```

```

end

```

```

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

```

```

lemmas eval-π = primes-pi-def[unfolded eval-prime-sum-upto]
lemmas eval-∅ = 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*]: $\pi \in \text{borel} \rightarrow_M \text{borel}$
and *measurable- ϑ* [*measurable*]: $\vartheta \in \text{borel} \rightarrow_M \text{borel}$
and *measurable- ψ* [*measurable*]: $\psi \in \text{borel} \rightarrow_M \text{borel}$
and *measurable- \mathfrak{M}* [*measurable*]: $\mathfrak{M} \in \text{borel} \rightarrow_M \text{borel}$
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*]: $x < 2 \implies \vartheta x = 0$
and *primes-M-eq-0* [*simp*]: $x < 2 \implies \mathfrak{M} x = 0$
unfolding *primes-pi-def* *primes-theta-def* *primes-M-def*
by (*rule prime-sum-upto-eq-0*; *simp*)**+**

lemma *π -nat-cancel* [*simp*]: $\pi (\text{nat } x) = \pi x$
and *ϑ -nat-cancel* [*simp*]: $\vartheta (\text{nat } x) = \vartheta x$
and *primes-M-nat-cancel* [*simp*]: $\mathfrak{M} (\text{nat } x) = \mathfrak{M} x$
and *ψ -nat-cancel* [*simp*]: $\psi (\text{nat } x) = \psi x$
and *π -floor-cancel* [*simp*]: $\pi (\text{of-int } \lfloor y \rfloor) = \pi y$
and *ϑ -floor-cancel* [*simp*]: $\vartheta (\text{of-int } \lfloor y \rfloor) = \vartheta y$
and *primes-M-floor-cancel* [*simp*]: $\mathfrak{M} (\text{of-int } \lfloor y \rfloor) = \mathfrak{M} y$
and *ψ -floor-cancel* [*simp*]: $\psi (\text{of-int } \lfloor y \rfloor) = \psi y$
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*]: $\vartheta x \geq 0$
and *primes-M-nonneg* [*intro*]: $\mathfrak{M} x \geq 0$
unfolding *primes-pi-def* *primes-theta-def* *primes-M-def*
by (*rule prime-sum-upto-nonneg*; *simp*)**+**

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

lemma *π -pos-iff*: $\pi x > 0 \iff x \geq 2$
proof
assume *x*: $x \geq 2$
show $\pi x > 0$
by (*rule less-le-trans[OF - π -mono[OF x]]*) (*auto simp: eval- π*)
next
assume $\pi x > 0$
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** $\text{nat } \lfloor x \rfloor \leq 1$ **by** *linarith*

hence *mangoldt* $n = (0 :: \text{real})$ **if** $n \in \{0 <.. \text{nat } \lfloor x \rfloor\}$ **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*]: $\psi x \geq 0$

unfolding *ψ -def sum-upto-def* **by** (*intro sum-nonneg mangoldt-nonneg*)

lemma *ψ -mono*: $x \leq y \implies \psi x \leq \psi y$

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* :: $\text{nat} \Rightarrow \text{nat}$ **where**

nth-prime $n = (\text{THE } p. \text{prime } p \wedge \text{card } \{q. \text{prime } q \wedge q < p\} = n)$

lemma *finite-primes-less* [*intro*]: *finite* $\{q::\text{nat}. \text{prime } q \wedge q < p\}$

by (*rule finite-subset[of - $\{..<p\}$]*) *auto*

lemma *nth-prime-unique-aux*:

fixes $p p' :: \text{nat}$

assumes *prime* p $\text{card } \{q. \text{prime } q \wedge q < p\} = n$

assumes *prime* p' $\text{card } \{q. \text{prime } q \wedge q < p'\} = n$

shows $p = p'$

using *assms*

proof (*induction p p' rule: linorder-wlog*)

case (*le p p'*)

have *finite* $\{q. \text{prime } q \wedge q < p'\}$ **by** (*rule finite-primes-less*)

moreover from *le* **have** $\{q. \text{prime } q \wedge q < p\} \subseteq \{q. \text{prime } q \wedge q < p'\}$

by *auto*

moreover from *le* **have** $\text{card } \{q. \text{prime } q \wedge q < p\} = \text{card } \{q. \text{prime } q \wedge q < p'\}$

by *simp*

ultimately have $\{q. \text{prime } q \wedge q < p\} = \{q. \text{prime } q \wedge q < p'\}$

by (*rule card-subset-eq*)

with $\langle \text{prime } p \rangle$ **have** $\neg(p < p')$ **by** *blast*

with $\langle p \leq p' \rangle$ **show** $p = p'$ **by** *auto*

qed auto

lemma π -smallest-prime-beyond:

$$\pi (\text{real } (\text{smallest-prime-beyond } m)) = \pi (\text{real } (m - 1)) + 1$$

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' \leq 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' = $\langle n < n' \rangle$ \langle prime n' \rangle this

have $\pi (\text{real } n') = \text{real } (\text{card } \{p. \text{prime } p \wedge p \leq n'\})$

by (simp add: π -def prime-sum-upto-def)

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

hence $\{p. \text{prime } p \wedge p \leq n'\} = \{p. \text{prime } p \wedge p \leq n\} \cup \{p. \text{prime } p \wedge p \in \{n < .. n'\}\}$

by auto

also have $\text{real } (\text{card } \dots) = \pi (\text{real } n) + \text{real } (\text{card } \{p. \text{prime } p \wedge p \in \{n < .. n'\}\})$

by (subst card-Un-disjoint) (auto simp: π -def prime-sum-upto-def)

also have $\{p. \text{prime } p \wedge p \in \{n < .. n'\}\} = \{n'\}$

using n' by (auto intro: antisym)

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

qed

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

proof (induction m)

case 0

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

next

case (Suc m)

from Suc.IH obtain n where n: $\pi (\text{real } n) = \text{real } m$

by auto

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

by (subst π -smallest-prime-beyond) auto

thus ?case by blast

qed

lemma nth-prime-exists: $\exists p::\text{nat}. \text{prime } p \wedge \text{card } \{q. \text{prime } q \wedge q < p\} = n$

proof –

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

hence card: $\text{card } \{q. \text{prime } q \wedge q \leq m\} = n$

by (*auto simp: π -def prime-sum-upto-def*)

define p **where** $p = \text{smallest-prime-beyond } (\text{Suc } m)$
have $m < p$ **using** *smallest-prime-beyond-le[of Suc m]* **unfolding** $p\text{-def}$ **by**
linarith
have *prime p* **by** (*simp add: p-def*)
have $p \leq q$ **if** *prime q* $q > m$ **for** q
using *smallest-prime-beyond-smallest[of q Suc m]* **that** **by** (*simp add: p-def*)
note $p = \langle m < p \rangle \langle \text{prime } p \rangle$ *this*

have $\{q. \text{prime } q \wedge q < p\} = \{q. \text{prime } q \wedge q \leq m\}$
proof *safe*
fix q **assume** *prime q* $q < p$
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** $\text{card } \dots = n$ **by** *fact*
finally **show** *?thesis* **using** $\langle \text{prime } p \rangle$ **by** *blast*
qed

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

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

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

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

lemma *nth-prime-eqI*:
assumes *prime p* $\text{card } \{q. \text{prime } q \wedge q < p\} = n$
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* $\text{card } \{q. \text{prime } q \wedge q \leq p\} = \text{Suc } n$
shows $\text{nth-prime } n = p$
proof (*rule nth-prime-eqI*)

have $\{q. \text{prime } q \wedge q \leq p\} = \text{insert } p \{q. \text{prime } q \wedge q < p\}$
using *assms* **by** *auto*
also have $\text{card } \dots = \text{Suc } (\text{card } \{q. \text{prime } q \wedge q < p\})$
by *simp*
finally show $\text{card } \{q. \text{prime } q \wedge q < p\} = n$ **using** *assms* **by** *simp*
qed (*use assms in auto*)

lemma *nth-prime-eqI''*:
assumes $\text{prime } p \ \pi \ (\text{real } p) = \text{real } n + 1$
shows $\text{nth-prime } n = p$
proof (*rule nth-prime-eqI'*)
have $\text{real } (\text{card } \{q. \text{prime } q \wedge q \leq p\}) = \pi \ (\text{real } p)$
by (*simp add: π -def prime-sum-upto-def*)
also have $\dots = \text{real } (\text{Suc } n)$ **by** (*simp add: assms*)
finally show $\text{card } \{q. \text{prime } q \wedge q \leq p\} = \text{Suc } n$
by (*simp only: of-nat-eq-iff*)
qed *fact+*

lemma *nth-prime-0* [*simp*]: $\text{nth-prime } 0 = 2$
by (*intro nth-prime-eqI*) (*auto dest: prime-gt-1-nat*)

lemma *nth-prime-Suc*: $\text{nth-prime } (\text{Suc } n) = \text{smallest-prime-beyond } (\text{Suc } (\text{nth-prime } n))$
by (*rule nth-prime-eqI''*) (*simp-all add: π -smallest-prime-beyond*)

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

lemma *strict-mono-nth-prime*: *strict-mono nth-prime*
proof (*rule strict-monoI-Suc*)
fix $n :: \text{nat}$
have $\text{Suc } (\text{nth-prime } n) \leq \text{smallest-prime-beyond } (\text{Suc } (\text{nth-prime } n))$ **by** *simp*
also have $\dots = \text{nth-prime } (\text{Suc } n)$ **by** (*simp add: nth-prime-Suc*)
finally show $\text{nth-prime } n < \text{nth-prime } (\text{Suc } n)$ **by** *simp*
qed

lemma *nth-prime-le-iff* [*simp*]: $\text{nth-prime } m \leq \text{nth-prime } n \longleftrightarrow m \leq n$
using *strict-mono-less-eq[OF strict-mono-nth-prime]* **by** *blast*

lemma *nth-prime-less-iff* [*simp*]: $\text{nth-prime } m < \text{nth-prime } n \longleftrightarrow m < n$
using *strict-mono-less[OF strict-mono-nth-prime]* **by** *blast*

lemma *nth-prime-eq-iff* [*simp*]: $\text{nth-prime } m = \text{nth-prime } n \longleftrightarrow m = n$
using *strict-mono-eq[OF strict-mono-nth-prime]* **by** *blast*

lemma *nth-prime-ge-2*: $\text{nth-prime } n \geq 2$
using *nth-prime-le-iff[of 0 n]* **by** (*simp del: nth-prime-le-iff*)

lemma *nth-prime-lower-bound*: $\text{nth-prime } n \geq \text{Suc } (\text{Suc } n)$
proof –

```

have  $n = \text{card } \{q. \text{prime } q \wedge q < \text{nth-prime } n\}$ 
  by simp
also have  $\dots \leq \text{card } \{2..<\text{nth-prime } n\}$ 
  by (intro card-mono) (auto dest: prime-gt-1-nat)
also have  $\dots = \text{nth-prime } n - 2$  by simp
finally show ?thesis using nth-prime-ge-2[of n] by linarith
qed

```

```

lemma nth-prime-at-top: filterlim nth-prime at-top at-top
proof (rule filterlim-at-top-mono)
  show filterlim ( $\lambda n::\text{nat}. n + 2$ ) at-top at-top by real-asymp
qed (auto simp: nth-prime-lower-bound)

```

```

lemma  $\pi$ -at-top: filterlim  $\pi$  at-top at-top
  unfolding filterlim-at-top
proof safe
  fix  $C :: \text{real}$ 
  define  $x0$  where  $x0 = \text{real } (\text{nth-prime } (\text{nat } \lceil \text{max } 0 \ C \rceil))$ 
  show eventually ( $\lambda x. \pi \ x \geq C$ ) at-top
    using eventually-ge-at-top
  proof eventually-elim
    fix  $x$  assume  $x \geq x0$ 
    have  $C \leq \text{real } (\text{nat } \lceil \text{max } 0 \ C \rceil + 1)$  by linarith
    also have  $\text{real } (\text{nat } \lceil \text{max } 0 \ C \rceil + 1) = \pi \ x0$ 
    unfolding x0-def by simp
    also have  $\dots \leq \pi \ x$  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 strict-mono-sequence-partition:
  assumes strict-mono ( $f :: \text{nat} \Rightarrow 'a :: \{\text{linorder}, \text{no-top}\}$ )
  assumes  $x \geq f \ 0$ 
  assumes filterlim f at-top at-top
  shows  $\exists k. x \in \{f \ k..<f \ (\text{Suc } k)\}$ 
proof -
  define  $k$  where  $k = (\text{LEAST } k. f \ (\text{Suc } k) > x)$ 
  {
    obtain  $n$  where  $x \leq f \ n$ 
    using assms by (auto simp: filterlim-at-top eventually-at-top-linorder)
    also have  $f \ n < f \ (\text{Suc } n)$ 
    using assms by (auto simp: strict-mono-Suc-iff)
    finally have  $\exists n. f \ (\text{Suc } n) > x$  by auto
  }
from LeastI-ex[OF this] have  $x < f \ (\text{Suc } k)$ 
  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 ?thesis by auto
qed

```

```

lemma nth-prime-partition:
  assumes  $x \geq 2$ 
  shows  $\exists k. x \in \{nth\text{-prime } k..<nth\text{-prime } (Suc k)\}$ 
  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  $\exists k. x \in \{real (nth\text{-prime } k)..<real (nth\text{-prime } (Suc k))\}$ 
  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  $n > nth\text{-prime } k$   $n < nth\text{-prime } (Suc k)$ 
  shows  $\neg prime n$ 
  using assms by (metis Suc-leI not-le nth-prime-Suc smallest-prime-beyond-smallest)

```

```

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

```

3.4 Relations between different prime-counting functions

The ψ function can be expressed as a sum of ϑ .

```

lemma  $\psi$ -altdef:
  assumes  $x > 0$ 

```


shows $\psi x = \text{sum-upto } (\lambda m. \text{prime-sum-upto } \ln (\text{root } m x)) (\log 2 x)$ (**is - =**
?rhs)

proof -

have *finite*: $\text{finite } \{p. \text{prime } p \wedge \text{real } p \leq y\}$ **for** y
by (*rule finite-subset*[*of - {..nat [y]}*]) (*auto simp: le-nat-iff' le-floor-iff*)

define S **where** $S = (\text{SIGMA } i:\{i. 0 < i \wedge \text{real } i \leq \log 2 x\}. \{p. \text{prime } p \wedge \text{real } p \leq \text{root } i x\})$

have $\psi x = (\sum (p, i) \mid \text{prime } p \wedge 0 < i \wedge \text{real } (p \wedge i) \leq x. \ln (\text{real } p))$ **unfolding**
 $\psi\text{-def}$

by (*subst sum-upto-primepows*[**where** $g = \lambda p i. \ln (\text{real } p)$])
(auto simp: case-prod-unfold mangoldt-non-primepow)

also have $\dots = (\sum (i, p) \mid \text{prime } p \wedge 0 < i \wedge \text{real } (p \wedge i) \leq x. \ln (\text{real } p))$

by (*intro sum.reindex-bij-witness*[*of - $\lambda(x,y). (y,x) \lambda(x,y). (y,x)$*]) *auto*

also have $\{(i, p). \text{prime } p \wedge 0 < i \wedge \text{real } (p \wedge i) \leq x\} = S$

unfolding $S\text{-def}$

proof safe

fix $i p :: \text{nat}$ **assume** $ip: i > 0 \text{ real } i \leq \log 2 x \text{ prime } p \text{ real } p \leq \text{root } i x$

hence $\text{real } (p \wedge i) \leq \text{root } i x \wedge i$ **unfolding** *of-nat-power* **by** (*intro power-mono*)

auto

with ip *assms* **show** $\text{real } (p \wedge i) \leq x$ **by** *simp*

next

fix $i p$ **assume** $ip: \text{prime } p \ i > 0 \text{ real } (p \wedge i) \leq x$

from ip **have** $2 \wedge i \leq p \wedge i$ **by** (*intro power-mono*) (*auto dest: prime-gt-1-nat*)

also have $\dots \leq x$ **using** ip **by** *simp*

finally show $\text{real } i \leq \log 2 x$

using *assms* **by** (*simp add: le-log-iff powr-realpow*)

have $\text{root } i (\text{real } p \wedge i) \leq \text{root } i x$ **using** ip *assms*

by (*subst real-root-le-iff*) *auto*

also have $\text{root } i (\text{real } p \wedge i) = \text{real } p$

using *assms ip* **by** (*subst real-root-pos2*) *auto*

finally show $\text{real } p \leq \text{root } i x$.

qed

also have $(\sum (i,p) \in S. \ln p) = \text{sum-upto } (\lambda m. \text{prime-sum-upto } \ln (\text{root } m x)) (\log 2 x)$

(log 2 x)

unfolding *sum-upto-def prime-sum-upto-def S-def* **using** *finite* **by** (*subst sum.Sigma*)

auto

finally show *?thesis* .

qed

lemma $\psi\text{-conv-}\vartheta\text{-sum}: x > 0 \implies \psi x = \text{sum-upto } (\lambda m. \vartheta (\text{root } m x)) (\log 2 x)$

by (*simp add: $\psi\text{-altdef } \vartheta\text{-def}$*)

lemma $\psi\text{-minus-}\vartheta$:

assumes $x: x \geq 2$

shows $\psi x - \vartheta x = (\sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 x. \vartheta (\text{root } i x))$

proof -

have *finite*: $\text{finite } \{i. 2 \leq i \wedge \text{real } i \leq \log 2 x\}$

by (*rule finite-subset*[*of - {2..nat [log 2 x]}*]) (*auto simp: le-nat-iff' le-floor-iff*)

have $\psi x = (\sum i \mid 0 < i \wedge \text{real } i \leq \log 2 x. \vartheta (\text{root } i x))$ **using** x

by (*simp add: ψ -conv- ϑ -sum sum-upto-def*)
 also have $\{i. 0 < i \wedge \text{real } i \leq \log 2 x\} = \text{insert } 1 \{i. 2 \leq i \wedge \text{real } i \leq \log 2 x\}$
 using x
 by (*auto simp: le-log-iff*)
 also have $(\sum i \in \dots \vartheta (\text{root } i x)) - \vartheta x =$
 $(\sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 x. \vartheta (\text{root } i x))$ using *finite*
 by (*subst sum.insert*) *auto*
 finally show *?thesis* .
 qed

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

lemma ϑ -conv- π -integral:

assumes $x \geq 2$
 shows $((\lambda t. \pi t / t) \text{ has-integral } (\pi x * \ln x - \vartheta x)) \{2..x\}$
proof (*cases $x = 2$*)
 case *False*
note [*intro*] = *finite-vimage-real-of-nat-greaterThanAtMost*
from *False* **and** *assms* **have** $x: x > 2$ **by** *simp*
have $((\lambda t. \text{sum-upto } (\text{ind prime}) t * (1 / t)) \text{ has-integral } \text{sum-upto } (\text{ind prime}) x * \ln x - \text{sum-upto } (\text{ind prime}) 2 * \ln 2 - (\sum n \in \text{real} -' \{2 <..x\}. \text{ind prime } n * \ln (\text{real } n))) \{2..x\}$ using x
by (*intro partial-summation-strong[where $X = \{\}$]*)
(auto intro!: continuous-intros derivative-eq-intros simp flip: has-field-derivative-iff-has-vector-derivative)
hence $((\lambda t. \pi t / t) \text{ has-integral } (\pi x * \ln x - (\pi 2 * \ln 2 + (\sum n \in \text{real} -' \{2 <..x\}. \text{ind prime } n * \ln n)))) \{2..x\}$
by (*simp add: π -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- π*)
also have $\dots = \vartheta x$ **unfolding** ϑ -def *prime-sum-upto-def* using x
by (*intro sum.mono-neutral-cong-right*) (*auto simp: ind-def dest: prime-gt-1-nat*)
 finally show *?thesis* .
 qed (*auto simp: has-integral-refl eval- π eval- ϑ*)

lemma π -conv- ϑ -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*] = *finite-vimage-real-of-nat-greaterThanAtMost*
from *False* **and** *assms* **have** $x: x > 2$ **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-field-derivative-iff-has-vector-derivative simp add: power2-eq-square)
also have $\text{sum-upto } b = \vartheta$
by (*simp add: ϑ -def b-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: b-def eval- ϑ*)
also have $(\sum n \in \text{insert } 2 (\text{real} - \{2 <..x\}). b n * (1 / \ln (\text{real } n))) = \pi x$ **using**
 x
unfolding π -def *prime-sum-upto-altdef1 sum-upto-def*
proof (*intro sum.mono-neutral-cong-left ballI, goal-cases*)
case ($\exists p$)
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- π eval- ϑ*)

lemma *integrable-weighted- ϑ :*
assumes $2 \leq a \leq x$
shows $((\lambda t. \vartheta t / (t * \ln t ^ 2)))$ *integrable-on* $\{a..x\}$
proof (*cases a < x*)
case *True*
hence $((\lambda t. \vartheta t * (1 / (t * \ln t ^ 2))))$ *integrable-on* $\{a..x\}$ **using** *assms*
unfolding ϑ -def *prime-sum-upto-altdef1*
by (*intro partial-summation-integrable-strong*[**where** $X = \{\}$ **and** $f = \lambda x. -1 / \ln x$])
(auto simp flip: has-field-derivative-iff-has-vector-derivative
intro!: derivative-eq-intros continuous-intros simp: power2-eq-square
field-simps)
thus *?thesis by simp*
qed (*insert has-integral-refl[of - a] assms, auto simp: has-integral-iff*)

lemma *ϑ -conv- \mathfrak{M} -integral:*
assumes $x \geq 2$
shows $(\mathfrak{M} \text{ has-integral } (\mathfrak{M} x * x - \vartheta x))$ $\{2..x\}$
proof (*cases x = 2*)
case *False*
with *assms have x: x > 2 by simp*
define $b :: \text{nat} \Rightarrow \text{real}$ **where** $b = (\lambda p. \text{ind prime } p * \ln p / p)$
note [*intro*] = *finite-vimage-real-of-nat-greaterThanAtMost*
have *prime-le-2: p = 2 if p ≤ 2 prime p for p :: nat*
using *that by (auto simp: prime-nat-iff)*

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

 $(\sum n \in \text{real} - \{2 <..x\}. b n * \text{real } n))$ $\{2..x\}$ **using** x
by (*intro partial-summation-strong*[*of* $\{\}$])

(auto simp flip: has-field-derivative-iff-has-vector-derivative
 intro!: derivative-eq-intros continuous-intros)
 also have sum-upto b = \mathfrak{M}
 by (simp add: fun-eq-iff primes-M-def b-def prime-sum-upto-altdef1)
 also have $\mathfrak{M} x * x - \mathfrak{M} 2 * 2 - (\sum n \in \text{real} - \{2 < .. x\}. b n * \text{real } n) =$
 $\mathfrak{M} x * x - (\sum n \in \text{insert } 2 (\text{real} - \{2 < .. x\}). b n * \text{real } n)$
 by (subst sum.insert) (auto simp: eval- \mathfrak{M} b-def)
 also have $(\sum n \in \text{insert } 2 (\text{real} - \{2 < .. x\}). b n * \text{real } n) = \vartheta x$
 unfolding ϑ -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 ?thesis by simp
 qed (auto simp: eval- ϑ eval- \mathfrak{M})

lemma \mathfrak{M} -conv- ϑ -integral:

assumes $x \geq 2$
 shows $((\lambda t. \vartheta t / t^2)$ has-integral $(\mathfrak{M} x - \vartheta x / x)$ $\{2..x\}$
 proof (cases $x = 2$)
 case False
 with assms have $x: x > 2$ 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 $p \leq 2$ 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
 $\text{sum-upto } b x * (-1 / x) - \text{sum-upto } b 2 * (-1 / 2) -$
 $(\sum n \in \text{real} - \{2 < .. x\}. b n * (-1 / \text{real } n))$ $\{2..x\}$ using x
 by (intro partial-summation-strong[of {}])
 (auto simp flip: has-field-derivative-iff-has-vector-derivative simp: power2-eq-square
 intro!: derivative-eq-intros continuous-intros)
 also have sum-upto b = ϑ
 by (simp add: fun-eq-iff ϑ -def b-def prime-sum-upto-altdef1)
 also have $\vartheta x * (-1 / x) - \vartheta 2 * (-1 / 2) - (\sum n \in \text{real} - \{2 < .. x\}. b n * (-1 / \text{real } n)) =$
 $-(\vartheta x / x - (\sum n \in \text{insert } 2 (\text{real} - \{2 < .. x\}). b n / \text{real } n))$
 by (subst sum.insert) (auto simp: eval- ϑ b-def sum-negf)
 also have $(\sum n \in \text{insert } 2 (\text{real} - \{2 < .. x\}). b n / \text{real } n) = \mathfrak{M} x$
 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 ?thesis by simp
 qed (auto simp: eval- ϑ eval- \mathfrak{M})

lemma integrable-primes-M: \mathfrak{M} integrable-on $\{x..y\}$ if $2 \leq x$ for $x y :: \text{real}$

proof -

have $(\lambda x. \mathfrak{M} x * 1)$ integrable-on $\{x..y\}$ if $2 \leq x$ $x < y$ 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-field-derivative-iff-has-vector-derivative
 intro!: derivative-eq-intros continuous-intros)

thus *?thesis* **using** *that has-integral-refl(2)*[of $\mathfrak{M} x$] **by** (*cases x y rule: linorder-cases*)
auto
qed

3.5 Bounds

lemma *ϑ -upper-bound-coarse:*

assumes $x \geq 1$

shows $\vartheta x \leq x * \ln x$

proof –

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

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

also have $\dots \leq \text{real-of-int } \lfloor x \rfloor * \ln x$ **using** *assms*

by (*simp add: sum-upto-altdef*)

also have $\dots \leq x * \ln x$ **using** *assms* **by** (*intro mult-right-mono*) *auto*

finally show *?thesis* .

qed

lemma *ϑ -le- ψ :* $\vartheta x \leq \psi x$

proof (*cases x \geq 2*)

case *False*

hence $\text{nat } \lfloor x \rfloor = 0 \vee \text{nat } \lfloor x \rfloor = 1$ **by** *linarith*

thus *?thesis* **by** (*auto simp: eval- ϑ*)

next

case *True*

hence $\psi x - \vartheta x = (\sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 x. \vartheta (\text{root } i x))$

by (*rule ψ -minus- ϑ*)

also have $\dots \geq 0$ **by** (*intro sum-nonneg*) *auto*

finally show *?thesis* **by** *simp*

qed

lemma *π -upper-bound-coarse:*

assumes $x \geq 0$

shows $\pi x \leq x / 3 + 2$

proof –

have $\{p. \text{prime } p \wedge p \leq \text{nat } \lfloor x \rfloor\} \subseteq \{2, 3\} \cup \{p. p \neq 1 \wedge \text{odd } p \wedge \neg 3 \text{ dvd } p \wedge p \leq \text{nat } \lfloor x \rfloor\}$

using *primes-dvd-imp-eq*[of 2 :: nat] *primes-dvd-imp-eq*[of 3 :: nat] **by** *auto*

also have $\dots \subseteq \{2, 3\} \cup ((\lambda k. 6*k+1) \cdot \{0 < .. < \text{nat } \lfloor (x+5)/6 \rfloor\} \cup (\lambda k. 6*k+5) \cdot \{.. < \text{nat } \lfloor (x+1)/6 \rfloor\})$

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

proof (*intro Un-mono subsetI*)

fix $p :: \text{nat}$ **assume** $p \in ?lhs$

hence $p: p \neq 1 \text{ odd } p \neg 3 \text{ dvd } p \wedge p \leq \text{nat } \lfloor x \rfloor$ **by** *auto*

from $p (1-3)$ **have** $(\exists k. k > 0 \wedge p = 6 * k + 1 \vee p = 6 * k + 5)$ **by**
presburger

then obtain k **where** $k > 0 \wedge p = 6 * k + 1 \vee p = 6 * k + 5$ **by** *blast*

hence $p = 6 * k + 1 \wedge k > 0 \wedge k < \text{nat } \lfloor (x+5)/6 \rfloor \vee p = 6*k+5 \wedge k < \text{nat}$

```

[(x+1)/6]
  unfolding add-divide-distrib using p(4) by linarith
  thus p ∈ ?rhs by auto
qed
finally have subset: {p. prime p ∧ p ≤ nat [x]} ⊆ ... (is - ⊆ ?A) .

have π x = real (card {p. prime p ∧ p ≤ nat [x]})
  by (simp add: π-def prime-sum-upto-altdef2)
also have card {p. prime p ∧ p ≤ nat [x]} ≤ card ?A
  by (intro card-mono subset) auto
also have ... ≤ 2 + (nat [(x+5)/6] - 1 + nat [(x+1)/6])
  by (intro order.trans[OF card-Un-le] add-mono order.trans[OF card-image-le])
auto
also have ... ≤ x / 3 + 2
  using assms unfolding add-divide-distrib by (cases x ≥ 1, linarith, simp)
finally show ?thesis by simp
qed

```

```

lemma le-numeral-iff: m ≤ numeral n ↔ m = numeral n ∨ m ≤ 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 ≡ (λn. (∏ {p::nat. prime p ∧ p ≤ n}))
  shows n > 0 ⇒ F n < 4 ^ n
proof (induction n rule: less-induct)
  case (less n)
  have n = 0 ∨ n = 1 ∨ n = 2 ∨ n = 3 ∨ even n ∧ n ≥ 4 ∨ odd n ∧ n ≥ 4
    by presburger
  then consider n = 0 | n = 1 | n = 2 | n = 3 | even n n ≥ 4 | odd n n ≥ 4
    by metis
  thus ?case
proof cases
  assume [simp]: n = 1
  have *: {p. prime p ∧ p ≤ Suc 0} = {} by (auto dest: prime-gt-1-nat)
  show ?thesis by (simp add: F-def *)
next
  assume [simp]: n = 2
  have *: {p. prime p ∧ p ≤ 2} = {2 :: nat}
    by (auto simp: le-numeral-iff dest: prime-gt-1-nat)
  thus ?thesis by (simp add: F-def *)
next
  assume [simp]: n = 3
  have *: {p. prime p ∧ p ≤ 3} = {2, 3 :: nat}
    by (auto simp: le-numeral-iff dest: prime-gt-1-nat)
  thus ?thesis by (simp add: F-def *)
next

```

```

assume n: even n n ≥ 4
from n have  $F (n - 1) < 4 \wedge (n - 1)$  by (intro less.IH) auto
also have  $\text{prime } p \wedge p \leq n \iff \text{prime } p \wedge p \leq n - 1$  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  $4 \wedge (n - 1) \leq (4 \wedge n :: \text{nat})$  by (intro power-increasing) auto
finally show ?case .
next
assume n: odd n n ≥ 4
then obtain k where k-eq:  $n = \text{Suc } (2 * k)$  by (auto elim: oddE)
from n have k:  $k \geq 2$  unfolding k-eq by presburger
have prime-dvd:  $p \text{ dvd } (n \text{ choose } k)$  if p:  $\text{prime } p \wedge p \in \{k+1 < .. n\}$  for p
proof -
  from p k n have  $p \text{ dvd pochhammer } (k + 2) k$ 
    unfolding pochhammer-prod
    by (subst prime-dvd-prod-iff)
      (auto intro!: beXI[of - p - k - 2] simp: k-eq numeral-2-eq-2 Suc-diff-Suc)
  also have  $\text{pochhammer } (\text{real } (k + 2)) k = \text{real } ((n \text{ choose } k) * \text{fact } k)$ 
    by (simp add: binomial-gbinomial gbinomial-pochhammer' k-eq field-simps)
  hence  $\text{pochhammer } (k + 2) k = (n \text{ choose } k) * \text{fact } k$ 
    unfolding pochhammer-of-nat of-nat-eq-iff .
  finally show  $p \text{ dvd } (n \text{ choose } k)$  using p
    by (auto simp: prime-dvd-fact-iff prime-dvd-mult-nat)
qed

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

```

qed
qed *auto*
hence $\prod \{p. \text{prime } p \wedge p \in \{k+1 \dots n\}\} \leq (n \text{ choose } k)$
by (*intro dvd-imp-le*) (*auto simp: k-eq*)
also have $\dots = 1 / 2 * (\sum i \in \{k, \text{Suc } k\}. n \text{ choose } i)$
using *central-binomial-odd*[of n] **by** (*simp add: k-eq*)
also have $(\sum i \in \{k, \text{Suc } k\}. n \text{ choose } i) < (\sum i \in \{0, k, \text{Suc } k\}. n \text{ choose } i)$
using k **by** *simp*
also have $\dots \leq (\sum i \leq n. n \text{ choose } i)$
by (*intro sum-mono2*) (*auto simp: k-eq*)
also have $\dots = (1 + 1) ^ n$
using *binomial*[of $1 1 n$] **by** *simp*
also have $1 / 2 * \dots = \text{real } (4 ^ k)$
by (*simp add: k-eq power-mult*)
finally have *less*: $(\prod \{p. \text{prime } p \wedge p \in \{k + 1 \dots n\}\}) < 4 ^ k$
unfolding *of-nat-less-iff* **by** *simp*

have $F n = F (\text{Suc } k) * (\prod \{p. \text{prime } p \wedge p \in \{k+1 \dots n\}\})$ **unfolding** *F-def*
by (*subst prod.union-disjoint* [*symmetric*]) (*auto intro!: prod.cong simp: k-eq*)
also have $\dots < 4 ^ \text{Suc } k * 4 ^ k$ **using** n
by (*intro mult-strict-mono less less.IH*) (*auto simp: k-eq*)
also have $\dots = 4 ^ (\text{Suc } k + k)$
by (*simp add: power-add*)
also have $\text{Suc } k + k = n$ **by** (*simp add: k-eq*)
finally show *?case* .

qed (*insert less.premis, auto*)
qed

lemma *∅-upper-bound*:
assumes $x: x \geq 1$
shows $\emptyset x < \ln 4 * x$
proof –
have $4 \text{ powr } (\emptyset x / \ln 4) = (\prod p \mid \text{prime } p \wedge p \leq \text{nat } \lfloor x \rfloor. 4 \text{ powr } (\log 4 (\text{real } p)))$
by (*simp add: ∅-def powr-sum prime-sum-upto-altdef2 sum-divide-distrib log-def*)
also have $\dots = (\prod p \mid \text{prime } p \wedge p \leq \text{nat } \lfloor x \rfloor. \text{real } p)$
by (*intro prod.cong*) (*auto dest: prime-gt-1-nat*)
also have $\dots = \text{real } (\prod p \mid \text{prime } p \wedge p \leq \text{nat } \lfloor x \rfloor. p)$
by *simp*
also have $(\prod p \mid \text{prime } p \wedge p \leq \text{nat } \lfloor x \rfloor. p) < 4 ^ \text{nat } \lfloor x \rfloor$
using x **by** (*intro prod-primes-upto-less*) *auto*
also have $\dots = 4 \text{ powr } \text{real } (\text{nat } \lfloor x \rfloor)$
using x **by** (*subst powr-realpow*) *auto*
also have $\dots \leq 4 \text{ powr } x$
using x **by** (*intro powr-mono*) *auto*
finally have $4 \text{ powr } (\emptyset x / \ln 4) < 4 \text{ powr } x$
by *simp*
thus $\emptyset x < \ln 4 * x$
by (*subst (asm) powr-less-cancel-iff*) (*auto simp: field-simps*)

qed

lemma ϑ -bigo: $\vartheta \in O(\lambda x. x)$

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 $x: x \geq 2$

shows $\psi x - \vartheta x \leq 2 * \ln x * \text{sqrt } x$

proof –

have $\psi x - \vartheta x = (\sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 x. \vartheta (\text{root } i x))$ **using** x
by (*rule* ψ -minus- ϑ)

also have $\dots \leq (\sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 x. \ln 4 * \text{root } i x)$

using x **by** (*intro sum-mono less-imp-le*[OF ϑ -upper-bound]) *auto*

also have $\dots \leq (\sum i \mid 2 \leq i \wedge \text{real } i \leq \log 2 x. \ln 4 * \text{root } 2 x)$ **using** x

by (*intro sum-mono mult-mono*) (*auto simp: le-log-iff powr-realpow intro!*:

real-root-decreasing)

also have $\dots = \text{card } \{i. 2 \leq i \wedge \text{real } i \leq \log 2 x\} * \ln 4 * \text{sqrt } x$

by (*simp add: sqrt-def*)

also have $\{i. 2 \leq i \wedge \text{real } i \leq \log 2 x\} = \{2..nat \lfloor \log 2 x \rfloor\}$

by (*auto simp: le-nat-iff' le-floor-iff*)

also have $\log 2 x \geq 1$ **using** x **by** (*simp add: le-log-iff*)

hence $\text{real } (nat \lfloor \log 2 x \rfloor - 1) \leq \log 2 x$ **using** x **by** *linarith*

hence $\text{card } \{2..nat \lfloor \log 2 x \rfloor\} \leq \log 2 x$ **by** *simp*

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

hence $\log 2 x * \ln 4 * \text{sqrt } x = 2 * \ln x * \text{sqrt } x$ **using** x

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

finally show *?thesis* **using** x **by** (*simp add: mult-right-mono*)

qed

lemma ψ -minus- ϑ -bigo: $(\lambda x. \psi x - \vartheta x) \in O(\lambda x. \ln x * \text{sqrt } x)$

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

fix $x :: \text{real}$ **assume** $x \geq 2$

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

using ψ -minus- ϑ -bound[*of* x] ϑ -le- ψ [*of* x] **by** *simp*

qed

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

proof –

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

by (*rule* ψ -minus- ϑ -bigo)

also have $(\lambda x. \ln x * \text{sqrt } x) \in O(\lambda x. x)$

by *real-asymp*

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

by (*rule sum-in-bigo*) (*fact* ϑ -bigo)

thus *?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 $\text{integral } \{2..x\} (\lambda t. 1 / \ln t^2) \leq \text{sqrt } x / \ln 2^2 + 4 * x / \ln x^2$

proof –

from x **have** $x * 1 \leq x^2$ **unfolding** *power2-eq-square* **by** (*intro mult-left-mono*) *auto*

with x **have** $x': 2 \leq \text{sqrt } x \text{ sqrt } x \leq x$

by (*auto simp: real-sqrt-le-iff' intro!: real-le-rsqrt*)

have $\text{integral } \{2..x\} (\lambda t. 1 / \ln t^2) =$

$\text{integral } \{2..\text{sqrt } x\} (\lambda t. 1 / \ln t^2) + \text{integral } \{\text{sqrt } x..x\} (\lambda t. 1 / \ln t^2)$

)

(**is** $= ?I1 + ?I2$) **using** $x x'$

by (*intro Henstock-Kurzweil-Integration.integral-combine [symmetric] integrable-continuous-real*) (*auto intro!: continuous-intros*)

also have $?I1 \leq \text{integral } \{2..\text{sqrt } x\} (\lambda t. 1 / \ln t^2)$ **using** x

by (*intro integral-le integrable-continuous-real divide-left-mono power-mono continuous-intros*) *auto*

also have $\dots \leq \text{sqrt } x / \ln 2^2$ **using** x' **by** (*simp add: field-simps*)

also have $?I2 \leq \text{integral } \{\text{sqrt } x..x\} (\lambda t. 1 / \ln (\text{sqrt } x)^2)$ **using** x'

by (*intro integral-le integrable-continuous-real divide-left-mono power-mono continuous-intros*) *auto*

also have $\dots \leq 4 * x / \ln x^2$ **using** x' **by** (*simp add: ln-sqrt field-simps*)

finally show *?thesis* **by** *simp*

qed

lemma *integral-one-over-log-squared-bigo*:

$(\lambda x::\text{real}. \text{integral } \{2..x\} (\lambda t. 1 / \ln t^2)) \in O(\lambda x. x / \ln x^2)$

proof –

define ub **where** $ub = (\lambda x::\text{real}. \text{sqrt } x / \ln 2^2 + 4 * x / \ln x^2)$

have *eventually* $(\lambda x. |\text{integral } \{2..x\} (\lambda t. 1 / (\ln t)^2)| \leq |ub x|)$ *at-top*

using *eventually-ge-at-top*[of 4]

proof *eventually-elim*

case (*elim* x)

hence $|\text{integral } \{2..x\} (\lambda t. 1 / \ln t^2)| = \text{integral } \{2..x\} (\lambda t. 1 / \ln t^2)$

by (*intro abs-of-nonneg integral-nonneg integrable-continuous-real continuous-intros*) *auto*

also have $\dots \leq |ub x|$

using *integral-one-over-log-squared-bound*[of x] *elim* **by** (*simp add: ub-def*)

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

lemma *π - ϑ -bound*:

assumes $x \geq (4 :: \text{real})$
defines $ub \equiv 2 / \ln 2 * \text{sqrt } x + 8 * \ln 2 * x / \ln x^2$
shows $\pi x - \vartheta x / \ln x \in \{0..ub\}$
proof –
define r **where** $r = (\lambda x. \text{integral } \{2..x\} (\lambda t. \vartheta t / (t * \ln t^2)))$
have *integrable*: $(\lambda t. c / \ln t^2)$ *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- ϑ [of 2 x] integrable[of ln 4] assms less-imp-le[OF ϑ -upper-bound]*
by *(intro integral-le divide-right-mono) (auto simp: field-simps)*
also have $\dots = \ln 4 * \text{integral } \{2..x\} (\lambda t. 1 / \ln t^2)$
using *integrable[of 1] by (subst integral-mult) auto*
also have $\dots \leq \ln 4 * (\text{sqrt } x / \ln 2^2 + 4 * x / \ln x^2)$
using *assms by (intro mult-left-mono integral-one-over-log-squared-bound) auto*
also have $\ln (4 :: \text{real}) = 2 * \ln 2$
using *ln-realpow[of 2 2] by simp*
also have $\dots * (\text{sqrt } x / \ln 2^2 + 4 * x / \ln x^2) = ub$
using *assms by (simp add: field-simps power2-eq-square ub-def)*
finally have $r x \leq \dots$.
moreover have $r x \geq 0$ **unfolding** *r-def* **using** *assms*
by *(intro integral-nonneg integrable-weighted- ϑ divide-nonneg-pos) auto*
ultimately have $r x \in \{0..ub\}$ **by** *auto*
with *π -conv- ϑ -integral[of x] assms(1)* **show** *?thesis*
by *(simp add: r-def has-integral-iff)*
qed

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

lemma *π - ϑ -bigO*: $(\lambda x. \pi x - \vartheta x / \ln x) \in O(\lambda x. x / \ln x^2)$

proof –

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

qed *auto*
also have $ub \in O(\lambda x. x / \ln x ^ 2)$
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 *π -upper-bound*:
assumes $x \geq (4 :: real)$
shows $\pi x < \ln 4 * x / \ln x + 8 * \ln 2 * x / \ln x ^ 2 + 2 / \ln 2 * \text{sqrt } x$
proof –
define *ub* **where** $ub = 2 / \ln 2 * \text{sqrt } x + 8 * \ln 2 * x / \ln x ^ 2$
have $\pi x \leq \vartheta x / \ln x + ub$
using *π - ϑ -bound[*of x*]* *assms* **unfolding** *ub-def* **by** *simp*
also from *assms* **have** $\vartheta x / \ln x < \ln 4 * x / \ln x$
by (*intro* *ϑ -upper-bound* *divide-strict-right-mono*) *auto*
finally show *?thesis*
using *assms* **by** (*simp add: algebra-simps ub-def*)
qed

lemma *π -bigo*: $\pi \in O(\lambda x. x / \ln x)$
proof –
have $(\lambda x. \pi x - \vartheta x / \ln x) \in O(\lambda x. x / \ln x ^ 2)$
by (*fact* *π - ϑ -bigo*)
also have $(\lambda x::real. x / \ln x ^ 2) \in O(\lambda x. x / \ln x)$
by *real-asymp*
finally have $(\lambda x. \pi x - \vartheta x / \ln x) \in O(\lambda x. x / \ln x)$.
moreover have *eventually* $(\lambda x::real. \ln x > 0)$ *at-top* **by** *real-asymp*
hence *eventually* $(\lambda x::real. \ln x \neq 0)$ *at-top* **by** *eventually-elim auto*
hence $(\lambda x. \vartheta x / \ln x) \in O(\lambda x. x / \ln x)$
using *ϑ -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 (*rule sum-in-bigo*)
thus *?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:

- $(1) \rightarrow (2) \rightarrow (3) \rightarrow (2) \rightarrow (1)$
- $(1) \rightarrow (4) \rightarrow (1)$
- $(4) \rightarrow (5) \rightarrow (4)$

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 *assms* **have** $((\lambda x. \pi x / (x / \ln x)) \longrightarrow 1) \text{ at-top}$

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

hence $((\lambda x. \ln(\pi x / (x / \ln x))) \longrightarrow \ln 1) \text{ at-top}$

by *(rule tendsto-ln) auto*

also have $?this \longleftrightarrow ((\lambda x. \ln(\pi x) - \ln x + \ln(\ln x)) \longrightarrow 0) \text{ at-top}$

by *(intro filterlim-cong eventually-mono[OF eventually-gt-at-top[of 2]])*
(auto simp: ln-div field-simps ln-mult π -pos)

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

by *(intro smalloI-tendsto) auto*

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 assms PNT1-imp-PNT1')*

also have $\dots \sim[at-top] (\lambda x. x)$

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 $(\lambda x. \pi x * \ln (\pi x)) \sim[at-top] (\lambda x. x)$

shows $nth\text{-prime} \sim[at-top] (\lambda n. n * \ln n)$

proof –

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

using *assms*

by $(rule\ asymp\equiv\ symI\ [OF\ asymp\equiv\ compose])$

$(auto\ intro!\ filterlim\ compose\ [OF\ filterlim\ real\ sequentially\ nth\text{-prime}\ at\ top])$

also have $\dots = (\lambda n. real (Suc\ n) * \ln (real (Suc\ n)))$

by $(simp\ add:\ add\ ac)$

also have $\dots \sim[at-top] (\lambda n. real\ n * \ln (real\ n))$

by *real-asymp*

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

qed

lemma *PNT3-imp-PNT2*:

assumes $nth\text{-prime} \sim[at-top] (\lambda n. n * \ln n)$

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

proof $(rule\ asymp\equiv\ symI, rule\ asymp\equiv\ sandwich\ real)$

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

at-top

using *eventually-ge-at-top*[of 2]

proof *eventually-elim*

case $(elim\ x)$

with *nth-prime-partition'*[of x] **show** *?case by auto*

qed

next

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

$(\lambda x. real (nat \lfloor \pi x \rfloor - 1) * \ln (real (nat \lfloor \pi x \rfloor - 1)))$

by $(rule\ asymp\equiv\ compose'\ [OF\ -\ \pi\ at\ top], rule\ asymp\equiv\ compose'\ [OF\ assms])\ real\ asymp$

also have $\dots \sim[at-top] (\lambda x. \pi x * \ln (\pi x))$

by $(rule\ asymp\equiv\ compose'\ [OF\ -\ \pi\ at\ top])\ real\ asymp$

finally show $(\lambda x. real (nth\text{-prime} (nat \lfloor \pi x \rfloor - 1))) \sim[at-top] (\lambda x. \pi x * \ln (\pi x))$.

next

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

$(\lambda x. real (nat \lfloor \pi x \rfloor) * \ln (real (nat \lfloor \pi x \rfloor)))$

by $(rule\ asymp\equiv\ compose'\ [OF\ -\ \pi\ at\ top], rule\ asymp\equiv\ compose'\ [OF\ assms])\ real\ asymp$

also have $\dots \sim[at-top] (\lambda x. \pi x * \ln (\pi x))$

by $(rule\ asymp\equiv\ compose'\ [OF\ -\ \pi\ at\ top])\ real\ asymp$

finally show $(\lambda x. real (nth\text{-prime} (nat \lfloor \pi x \rfloor))) \sim[at-top] (\lambda x. \pi x * \ln (\pi x))$.

qed

lemma *PNT2-imp-PNT1*:

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

shows $(\lambda x. \ln (\pi x)) \sim[at-top] (\lambda x. \ln x)$

and $\pi \sim[at-top] (\lambda x. x / \ln x)$
proof –
have ev : *eventually* $(\lambda x. \pi x > 0)$ *at-top*
eventually $(\lambda x. \ln (\pi x) > 0)$ *at-top*
eventually $(\lambda x. \ln (\ln (\pi x)) > 0)$ *at-top*
by (*rule eventually-compose-filterlim*[*OF* - π -*at-top*], *real-asymp*) +

let $?f = \lambda x. 1 + \ln (\ln (\pi x)) / \ln (\pi x) - \ln x / \ln (\pi x)$
have $((\lambda x. \ln (\pi x) * ?f x) \longrightarrow \ln 1)$ *at-top*
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 $\forall_F x$ *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-mult* *ln-div*)
qed
moreover have $((\lambda x. 1 / \ln (\pi x)) \longrightarrow 0)$ *at-top*
by (*rule filterlim-compose*[*OF* - π -*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-mult*)
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* - π -*at-top*]) (*real-asymp* | *simp*) +
hence $((\lambda x. \ln x / \ln (\pi x)) \longrightarrow 1)$ *at-top*
by *simp*
thus $*$: $(\lambda x. \ln (\pi x)) \sim[at-top] (\lambda x. \ln x)$
by (*rule asymp-equiv-symI*[*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 $\pi \sim[at-top] (\lambda x. \pi x * \ln (\pi x) / \ln (\pi x))$
by (*rule asymp-equiv-refl-ev*)
also from *assms* **and** $*$ **have** $(\lambda x. \pi x * \ln (\pi x) / \ln (\pi x)) \sim[at-top] (\lambda x. x / \ln x)$
by (*rule asymp-equiv-intros*)
finally show $\pi \sim[at-top] (\lambda x. x / \ln x)$.
qed

lemma *PNT4-imp-PNT5*:
assumes $\vartheta \sim[at-top] (\lambda x. x)$
shows $\psi \sim[at-top] (\lambda x. x)$
proof –
define r **where** $r = (\lambda x. \psi x - \vartheta x)$

have $r \in O(\lambda x. \ln x * \text{sqrt } x)$
unfolding $r\text{-def}$ **by** (*fact $\psi\text{-minus-}\vartheta\text{-bigo}$*)
also have $(\lambda x::\text{real}. \ln x * \text{sqrt } x) \in o(\lambda x. x)$
by *real-asymp*
finally have $r: r \in o(\lambda x. x)$.

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

qed

lemma *PNT4-imp-PNT1*:

assumes $\vartheta \sim[at\text{-top}] (\lambda x. x)$
shows $\pi \sim[at\text{-top}] (\lambda x. x / \ln x)$

proof –

have $(\lambda x. (\pi x - \vartheta x / \ln x) + ((\vartheta x - x) / \ln x)) \in o(\lambda x. x / \ln x)$

proof (*rule sum-in-smallo*)

have $(\lambda x. \pi x - \vartheta x / \ln x) \in O(\lambda x. x / \ln x \wedge 2)$

by (*rule $\pi\text{-}\vartheta\text{-bigo}$*)

also have $(\lambda x. x / \ln x \wedge 2) \in o(\lambda x. x / \ln x :: \text{real})$

by *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** *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 asymp-equiv-imp-diff-smallo assms*)

qed

thus *?thesis* **by** (*simp add: diff-divide-distrib asymp-equiv-altdef*)

qed

lemma *PNT1-imp-PNT4*:

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

shows $\vartheta \sim[at\text{-top}] (\lambda x. x)$

proof –

have $\vartheta \sim[at\text{-top}] (\lambda x. \pi x * \ln x)$

proof (*rule smallo-imp-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 bighetaI-cong eventually-mono[OF eventually-gt-at-top[of 1]]*)

(*auto simp: 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 *landau-o.big.uminus-in-iff* **by** (*intro landau-o.big.mult-right $\pi\text{-}\vartheta\text{-bigo}$*)

also have $(\lambda x::\text{real}. x / (\ln x)^2 * \ln x) \in o(\lambda x. x / \ln x * \ln x)$

by *real-asymp*

also have $(\lambda x. x / \ln x * \ln x) \in \Theta(\lambda x. \pi x * \ln x)$

by (*intro asymp-equiv-imp-bigheta asymp-equiv-intros asymp-equiv-symI[OF assms]*)

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

qed

also have $\dots \sim[at-top] (\lambda x. x / \ln x * \ln x)$
by (*intro asymp-equiv-intros assms*)
also have $\dots \sim[at-top] (\lambda x. x)$
by *real-asymp*
finally show *?thesis* .
qed

lemma *PNT5-imp-PNT4*:
assumes $\psi \sim[at-top] (\lambda x. x)$
shows $\vartheta \sim[at-top] (\lambda x. x)$
proof –
define r **where** $r = (\lambda x. \vartheta x - \psi x)$
have $(\lambda x. \psi x - \vartheta x) \in O(\lambda x. \ln x * \sqrt{x})$
by (*fact psi-minus- ϑ -bigO*)
also have $(\lambda x. \psi x - \vartheta x) = (\lambda x. -r x)$
by (*simp add: r-def*)
finally have $r \in O(\lambda x. \ln x * \sqrt{x})$
by *simp*
also have $(\lambda x::real. \ln x * \sqrt{x}) \in o(\lambda x. x)$
by *real-asymp*
finally have $r: r \in o(\lambda x. x)$.

have $(\lambda x. \psi x + r x) \sim[at-top] (\lambda x. x)$
using *assms r* **by** (*subst asymp-equiv-add-right auto*)
thus *?thesis* **by** (*simp add: r-def*)
qed

3.7 The asymptotic form of Mertens' First Theorem

Mertens' first theorem states that $\mathfrak{M}(x) - \ln x$ is bounded, i. e. $\mathfrak{M}(x) = \ln x + O(1)$.

With some work, one can also show some absolute bounds for $|\mathfrak{M}(x) - \ln x|$, and we will, in fact, do this later. However, this asymptotic form is somewhat easier to obtain and it is (as we shall see) enough to prove the Prime Number Theorem, so we prove the weak form here first for the sake of a smoother presentation.

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

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

We show this using summation by parts.

lemma *stirling-weak*:
assumes $x: x \geq 1$
shows $\text{sum-upto } \ln x \in \{x * \ln x - x - \ln x + 1 .. x * \ln x\}$
proof (*cases x = 1*)
case *True*

have $\{0 < \dots \text{Suc } 0\} = \{1\}$ **by** *auto*
with *True* **show** *?thesis* **by** (*simp add: sum-upto-altdef*)
next
case *False*
with *assms* **have** $x: x > 1$ **by** *simp*
have $((\lambda t. \text{sum-upto } (\lambda \cdot. 1) t * (1 / t)) \text{ has-integral } \text{sum-upto } (\lambda \cdot. 1) x * \ln x - \text{sum-upto } (\lambda \cdot. 1) 1 * \ln 1 - (\sum_{n \in \text{real} - \{1 < \cdot x\}} 1 * \ln (\text{real } n))) \{1..x\}$ **using** x
by (*intro partial-summation-strong[of {}]*)
(auto simp flip: has-field-derivative-iff-has-vector-derivative intro!: derivative-eq-intros continuous-intros)
hence $((\lambda t. \text{real } (\text{nat } \lfloor t \rfloor) / t) \text{ has-integral } \text{real } (\text{nat } \lfloor x \rfloor) * \ln x - (\sum_{n \in \text{real} - \{1 < \cdot x\}} \ln (\text{real } n))) \{1..x\}$
by (*simp add: sum-upto-altdef*)
also have $(\sum_{n \in \text{real} - \{1 < \cdot x\}} \ln (\text{real } n)) = \text{sum-upto } \ln x$ **unfolding** *sum-upto-def*
by (*intro sum.mono-neutral-left*)
(auto intro!: finite-subset[OF finite-vimage-real-of-nat-greaterThanAtMost[of 0 x]])
finally have $*$: $((\lambda t. \text{real } (\text{nat } \lfloor t \rfloor) / t) \text{ has-integral } \lfloor x \rfloor * \ln x - \text{sum-upto } \ln x) \{1..x\}$
using x **by** *simp*

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

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

lemma *stirling-weak-bigo*: $(\lambda x::\text{real}. \text{sum-upto } \ln x - x * \ln x) \in O(\lambda x. x)$
proof –
have $(\lambda x. \text{sum-upto } \ln x - x * \ln x) \in O(\lambda x. -(\text{sum-upto } \ln x - x * \ln x))$
by (*subst landau-o.big.uminus*) *auto*
also have $(\lambda x. -(\text{sum-upto } \ln x - x * \ln x)) \in O(\lambda x. x + \ln x - 1)$
proof (*intro le-imp-bigo-real[of 2] eventually-mono[OF eventually-ge-at-top[of 1]], goal-cases*)

```

    case (2 x)
    thus ?case using stirling-weak[of x] by (auto simp: algebra-simps)
next
    case (3 x)
    thus ?case using stirling-weak[of x] by (auto simp: algebra-simps)
qed auto
also have  $(\lambda x. x + \ln x - 1) \in O(\lambda x::\text{real}. x)$  by real-asymp
finally show ?thesis .
qed

```

```

lemma floor-floor-div-eq:
  fixes  $x :: \text{real}$  and  $d :: \text{nat}$ 
  assumes  $x \geq 0$ 
  shows  $\lfloor \text{nat } \lfloor x \rfloor / \text{real } d \rfloor = \lfloor x / \text{real } d \rfloor$ 
proof -
  have  $\lfloor \text{nat } \lfloor x \rfloor / \text{real-of-int } (\text{int } d) \rfloor = \lfloor x / \text{real-of-int } (\text{int } d) \rfloor$  using assms
  by (subst (1 2) floor-divide-real-eq-div) auto
  thus ?thesis by simp
qed

```

The key to showing Mertens' first theorem is the function

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

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

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

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

```

proof -
  define f where  $f = (\lambda d. \text{mangoldt } d / \text{real } d)$ 
  define C where  $C = (\sum p. \ln (\text{real } (p + 1)) * (1 / \text{real } (p * (p - 1))))$ 
  have summable: summable  $(\lambda p::\text{nat}. \ln (p + 1) * (1 / (p * (p - 1))))$ 
  proof (rule summable-comparison-test-bigo)
    show summable  $(\lambda p. \text{norm } (p \text{ powr } (-3/2)))$ 
    by (simp add: summable-real-powr-iff)
  qed real-asymp

```

have diff-bound: $\text{sum-upto } f x - \mathfrak{M} x \in \{0..C\}$ if $x: x \geq 4$ for x

```

proof -
  define S where  $S = \{(p, i). \text{prime } p \wedge 0 < i \wedge \text{real } (p \wedge i) \leq x\}$ 
  define S' where  $S' = (\text{SIGMA } p:\{2.. \text{nat } \lfloor \text{root } 2 x \rfloor\}. \{2.. \text{nat } \lfloor \log 2 x \rfloor\})$ 
  have  $S \subseteq \{.. \text{nat } \lfloor x \rfloor\} \times \{.. \text{nat } \lfloor \log 2 x \rfloor\}$  unfolding S-def
    using x primepows-le-subset[of x 1] by (auto simp: Suc-le-eq)
  hence finite S by (rule finite-subset) auto
  note fn = finite-subset[OF - this, unfolded S-def]

```

have $\text{sum-upto } f x = (\sum (p, i) \in S. \ln (\text{real } p) / \text{real } (p \wedge i))$ unfolding S-def

by (intro sum-upto-primewows) (auto simp: f-def mangoldt-non-primewow)
 also have $S = \{p. \text{prime } p \wedge p \leq x\} \times \{1\} \cup \{(p, i). \text{prime } p \wedge 1 < i \wedge \text{real } (p \wedge i) \leq x\}$
 by (auto simp: S-def not-less le-Suc-eq not-le intro!: Suc-lessI)
 also have $(\sum (p, i) \in \dots \ln (\text{real } p) / \text{real } (p \wedge i)) =$
 $(\sum (p, i) \in \{p. \text{prime } p \wedge \text{of-nat } p \leq x\} \times \{1\}. \ln (\text{real } p) / \text{real } (p \wedge i)) +$
 $(\sum (p, i) \mid \text{prime } p \wedge \text{real } (p \wedge i) \leq x \wedge i > 1. \ln (\text{real } p) / \text{real } (p \wedge i))$
 (is - = ?S1 + ?S2)
 by (subst sum.union-disjoint[OF fin fin]) (auto simp: conj-commute case-prod-unfold)
 also have ?S1 = $\mathfrak{M} x$
 by (subst sum.cartesian-product [symmetric]) (auto simp: primes-M-def prime-sum-upto-def)
 finally have eq: sum-upto f x - $\mathfrak{M} x = ?S2$ by simp
 have ?S2 $\leq (\sum (p, i) \in S'. \ln (\text{real } p) / \text{real } (p \wedge i))$
 using primewows-le-subset[of x 2] x unfolding case-prod-unfold of-nat-power
 by (intro sum-mono2 divide-nonneg-pos zero-less-power)
 (auto simp: eval-nat-numeral Suc-le-eq S'-def subset-iff dest: prime-gt-1-nat)+
 also have $\dots = (\sum p=2..nat \lfloor \text{sqrt } x \rfloor. \ln p * (\sum i \in \{2..nat \lfloor \log 2 x \rfloor\}. (1 / \text{real } p) \wedge i))$
 by (simp add: S'-def sum.Sigma case-prod-unfold sum-distrib-left sqrt-def field-simps)
 also have $\dots \leq (\sum p=2..nat \lfloor \text{sqrt } x \rfloor. \ln p * (1 / (p * (p - 1))))$
 unfolding sum-upto-def
 proof (intro sum-mono, goal-cases)
 case (1 p)
 from x have $\text{nat } \lfloor \log 2 x \rfloor \geq 2$
 by (auto simp: le-nat-iff' le-log-iff)
 hence $(\sum i \in \{2..nat \lfloor \log 2 x \rfloor\}. (1 / \text{real } p) \wedge i) =$
 $((1 / p)^2 - (1 / p) \wedge \text{nat } \lfloor \log 2 x \rfloor / p) / (1 - 1 / p)$ using 1
 by (subst sum-gp) (auto dest!: prime-gt-1-nat simp: field-simps power2-eq-square)
 also have $\dots \leq ((1 / p) \wedge 2 - 0) / (1 - 1 / p)$
 using 1 by (intro divide-right-mono diff-mono power-mono)
 (auto simp: field-simps dest: prime-gt-0-nat)
 also have $\dots = 1 / (p * (p - 1))$
 by (auto simp: divide-simps power2-eq-square dest: prime-gt-0-nat)
 finally show ?case
 using 1 by (intro mult-left-mono) (auto dest: prime-gt-0-nat)
 qed
 also have $\dots \leq (\sum p=2..nat \lfloor \text{sqrt } x \rfloor. \ln (p + 1) * (1 / (p * (p - 1))))$
 by (intro sum-mono mult-mono) auto
 also have $\dots \leq C$ unfolding C-def
 by (intro sum-le-suminf summable) auto
 finally have ?S2 $\leq C$ by simp
 moreover have ?S2 ≥ 0 by (intro sum-nonneg) (auto dest: prime-gt-0-nat)
 ultimately show ?thesis using eq by simp
 qed
 from diff-bound[of 4] have $C \geq 0$ by auto

with *diff-bound* **show** $(\lambda x. \text{sum-upto } f x - \mathfrak{M} x) \in O(\lambda-. 1)$
by (*intro le-imp-bigo-real*[of C] *eventually-mono*[OF *eventually-ge-at-top*[of 4]])
auto
qed

Next, we show that our $h(x)$ itself is close to $\ln x$, i. e.:

$$\sum_{n \leq x} \frac{\Lambda(d)}{d} = \ln x + O(1)$$

lemma *sum-upto-mangoldt-over-id-asymptotics*:

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

proof –

define r **where** $r = (\lambda n::\text{real}. \text{sum-upto } (\lambda d. \text{mangoldt } d * (n / d - \text{real-of-int } \lfloor n / d \rfloor)) n)$

have $r: r \in O(\psi)$

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

fix $x :: \text{real}$ **assume** $x \geq 0$

have $\text{eq}: \{1..nat \lfloor x \rfloor\} = \{0<..nat \lfloor x \rfloor\}$ **by** *auto*

hence $r x \geq 0$ **unfolding** $r\text{-def}$ *sum-upto-def*

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

(*auto simp: floor-le-iff*)

moreover **have** $x / \text{real } d \leq 1 + \text{real-of-int } \lfloor x / \text{real } d \rfloor$ **for** d **by** *linarith*

hence $r x \leq \text{sum-upto } (\lambda d. \text{mangoldt } d * 1) x$ **unfolding** *sum-upto-altdef* eq $r\text{-def}$ **using** x

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

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

ultimately **show** $\text{norm } (r x) \leq 1 * \text{norm } (\psi x)$ **by** (*simp add: psi-def*)

qed *auto*

also **have** $\psi \in O(\lambda x. x)$ **by** (*fact psi-bigo*)

finally **have** $r: r \in O(\lambda x. x)$.

define r' **where** $r' = (\lambda x::\text{real}. \text{sum-upto } \ln x - x * \ln x)$

have $r'\text{-bigo}: r' \in O(\lambda x. x)$

using *stirling-weak-bigo* **unfolding** $r'\text{-def}$.

have $\text{ln-fact}: \ln (\text{fact } n) = (\sum d=1..n. \ln d)$ **for** n

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

hence $r': \text{sum-upto } \ln n = n * \ln n + r' n$ **for** $n :: \text{real}$

unfolding $r'\text{-def}$ *sum-upto-altdef* **by** (*auto intro!: sum.cong*)

have *eventually* $(\lambda n. \text{sum-upto } (\lambda d. \text{mangoldt } d / d) n - \ln n = r' n / n + r n / n)$ *at-top*

using *eventually-gt-at-top*

proof *eventually-elim*

fix $x :: \text{real}$ **assume** $x > 0$

have $\text{sum-upto } \ln x = \text{sum-upto } (\lambda n. \text{mangoldt } n * \text{real } (\text{nat } \lfloor x / n \rfloor)) x$

unfolding *sum-upto-ln-conv-sum-upto-mangoldt* ..

also **have** ... = $\text{sum-upto } (\lambda d. \text{mangoldt } d * (x / d)) x - r x$

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

also have $\text{sum-upto } (\lambda d. \text{mangoldt } d * (x / d)) x = x * \text{sum-upto } (\lambda d. \text{mangoldt } d / d) x$
unfolding sum-upto-def **by** (*subst sum-distrib-left*) (*simp add: field-simps*)
finally have $x * \text{sum-upto } (\lambda d. \text{mangoldt } d / \text{real } d) x = r' x + r x + x * \ln x$
by (*simp add: r' algebra-simps*)
thus $\text{sum-upto } (\lambda d. \text{mangoldt } d / d) x - \ln x = r' x / x + r x / x$
using x **by** (*simp add: field-simps*)
qed
hence $(\lambda x. \text{sum-upto } (\lambda d. \text{mangoldt } d / d) x - \ln x) \in \Theta(\lambda x. r' x / x + r x / x)$
by (*rule bigthetaI-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*: $(\lambda x. \mathfrak{M} x - \ln x) \in O(\lambda-. 1)$
proof –
define f **where** $f = \text{sum-upto } (\lambda d. \text{mangoldt } d / d)$
have $(\lambda x. (f x - \ln x) - (f x - \mathfrak{M} x)) \in O(\lambda-. 1)$
using *sum-upto-mangoldt-over-id-asymptotics*
sum-upto-mangoldt-over-id-minus-phi-bounded
unfolding $f\text{-def}$ **by** (*rule sum-in-bigo*)
thus *?thesis* **by** *simp*
qed

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

end

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. $\mathfrak{M}(x) = \ln x + O(1)$, we now want to derive that $\mathfrak{M}(x) = \ln x + c + o(1)$. This result is considerably stronger and it implies the Prime Number Theorem quite directly.

In order to do this, we define the Dirichlet series

$$f(s) = \sum_{n=1}^{\infty} \frac{\mathfrak{M}(n)}{n^s} .$$

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 *fds-newman where*

fds-newman = fds (λn. complex-of-real (ℳ n))

lemma *fds-nth-newman:*

fds-nth fds-newman n = of-real (ℳ n)

by (*simp add: fds-newman-def fds-nth-fds*)

lemma *norm-fds-nth-newman:*

norm (fds-nth fds-newman n) = ℳ n

unfolding *fds-nth-newman norm-of-real*

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 $\mathfrak{M}(n) - \ln n$, so by Mertens' first theorem, $f(s) + \zeta'(s)$ has bounded coefficients.

lemma *bounded-coeffs-newman-minus-deriv-zeta:*

defines *f ≡ fds-newman + fds-deriv fds-zeta*

shows *Bseq (λn. fds-nth f n)*

proof –

have (*λn. ℳ (real n) - ln (real n) ∈ O(λ-. 1)*)

using *mertens-bounded by (rule landau-o.big.compose) real-asymp*

from *natfun-bigo-1E[OF this, of 1]*

obtain *c where c: c ≥ 1 ∧ n. |ℳ (real n) - ln (real n)| ≤ c by auto*

show *?thesis*

proof (*intro BseqI[of c] allI*)

fix *n :: nat*

show *norm (fds-nth f n) ≤ c*

proof (*cases n = 0*)

case *False*

hence *fds-nth f n = of-real (ℳ n - ln n)*

by (*simp add: f-def fds-nth-newman fds-nth-deriv fds-nth-zeta scaleR-conv-of-real*)

also from *⟨n ≠ 0⟩ have norm ... ≤ c*

using *c(2)[of n] by (simp add: in-Reals-norm)*

finally show *?thesis .*

qed (*insert c, auto*)

qed (*insert c, auto*)

qed

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

lemma *abs-conv-abscissa-newman*: *abs-conv-abscissa fds-newman* ≤ 1
and *conv-abscissa-newman*: *conv-abscissa fds-newman* ≤ 1

proof –

define f **where** $f = \text{fds-newman} + \text{fds-deriv fds-zeta}$
have *abs-conv-abscissa* $f \leq 1$
using *bounded-coeffs-newman-minus-deriv-zeta unfolding f-def*
by (*rule bounded-coeffs-imp-abs-conv-abscissa-le-1*)
hence *abs-conv-abscissa* $(f - \text{fds-deriv fds-zeta}) \leq 1$
by (*intro abs-conv-abscissa-diff-leI*) (*auto simp: abs-conv-abscissa-deriv*)
also have $f - \text{fds-deriv fds-zeta} = \text{fds-newman}$ **by** (*simp add: f-def*)
finally show *abs-conv-abscissa fds-newman* ≤ 1 .
from *conv-le-abs-conv-abscissa* **and this show** *conv-abscissa fds-newman* ≤ 1
by (*rule order.trans*)

qed

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

lemma *eval-fds-newman-conv-infsetsum*:

assumes $s: \text{Re } s > 1$

shows *eval-fds-newman* $s = (\sum_{a \mid p} \ln(\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \text{ } s$
 $(\lambda p. \ln(\text{real } p) / \text{real } p * \text{hurwitz-zeta } p \text{ } s)$ *abs-summable-on* $\{p. \text{prime } p\}$

proof –

from s **have** *conv: fds-abs-converges fds-newman* s
by (*intro fds-abs-converges le-less-trans[OF abs-conv-abscissa-newman]*) *auto*
define f **where** $f = (\lambda n p. \ln(\text{real } p) / \text{real } p / \text{of-nat } n \text{ } \text{powr } s)$

have *eq*: $(\sum_{a \mid n} f \text{ } n \text{ } p) = \ln(\text{real } p) / \text{real } p * \text{hurwitz-zeta } p \text{ } s$ **if** *prime* p
for p

proof –

have $(\sum_{a \mid n} f \text{ } n \text{ } p) = (\sum_{a \mid n} \ln(\text{real } p) / \text{of-nat } p) * (1 / \text{of-nat } x \text{ } \text{powr } s)$

by (*simp add: f-def*)

also have $\dots = (\ln(\text{real } p) / \text{of-nat } p) * (\sum_{a \mid n} 1 / \text{of-nat } x \text{ } \text{powr } s)$

using *abs-summable-hurwitz-zeta[of s 0 p]* *that* s

by (*intro infsetsum-cmult-right*) (*auto dest: prime-gt-0-nat*)

also have $(\sum_{a \mid n} 1 / \text{of-nat } x \text{ } \text{powr } s) = \text{hurwitz-zeta } p \text{ } s$

using s *that* **by** (*subst hurwitz-zeta-nat-conv-infsetsum(2)*)

(*auto dest: prime-gt-0-nat simp: field-simps powr-minus*)

finally show *?thesis* .

qed

have *norm-f*: *norm* $(f \text{ } n \text{ } p) = \ln p / p / n \text{ } \text{powr } \text{Re } s$ **if** *prime* p **for** $n :: \text{nat}$

by (*auto simp: f-def norm-divide norm-mult norm-powr-real-powr*)

from *conv* **have** $(\lambda n. \text{norm } (\text{fds-nth } \text{fds-newman } n / n \text{ powr } s)) \text{ abs-summable-on } UNIV$
by (*intro abs-summable-on-normI*) (*simp add: fds-abs-converges-altdef'*)
also have $(\lambda n. \text{norm } (\text{fds-nth } \text{fds-newman } n / n \text{ powr } s)) =$
 $(\lambda n. \sum p \mid \text{prime } p \wedge p \leq n. \text{norm } (f \ n \ p))$
by (*auto simp: norm-divide norm-fds-nth-newman sum-divide-distrib primes-M-def*
prime-sum-upto-def norm-mult norm-f norm-powr-real-powr intro!
sum.cong)
finally have *summable1*: $(\lambda(n,p). f \ n \ p) \text{ abs-summable-on } (SIGMA \ n:UNIV. \{p. \text{prime } p \wedge p \leq n\})$
using *conv* **by** (*subst abs-summable-on-Sigma-iff*) *auto*
also have *?this* $\longleftrightarrow (\lambda(p,n). f \ n \ p) \text{ abs-summable-on}$
 $(\lambda(n,p). (p,n)) \text{ ' } (SIGMA \ n:UNIV. \{p. \text{prime } p \wedge p \leq n\})$
by (*subst abs-summable-on-reindex-iff [symmetric]*) (*auto simp: case-prod-unfold*
inj-on-def)
also have $(\lambda(n,p). (p,n)) \text{ ' } (SIGMA \ n:UNIV. \{p. \text{prime } p \wedge p \leq n\}) =$
 $(SIGMA \ p:\{p. \text{prime } p\}. \{p.. \}) \text{ by } auto$
finally have *summable2*: $(\lambda(p,n). f \ n \ p) \text{ abs-summable-on } \dots$
from *abs-summable-on-Sigma-project1* [*OF this*]
have $(\lambda p. \sum_{a \in \{p.. \}} f \ n \ p) \text{ abs-summable-on } \{p. \text{prime } p\} \text{ by } auto$
also have *?this* $\longleftrightarrow (\lambda p. \ln (\text{real } p) / \text{real } p * \text{hurwitz-zeta } p \ s) \text{ abs-summable-on}$
 $\{p. \text{prime } p\}$
by (*intro abs-summable-on-cong eq*) *auto*
finally show \dots .

have *eval-fds fds-newman s* =
 $(\sum_{a \in n}. \sum p \mid \text{prime } p \wedge p \leq n. \ln (\text{real } p) / \text{real } p / \text{of-nat } n \text{ powr } s)$
using *conv* **by** (*simp add: eval-fds-altdef fds-nth-newman sum-divide-distrib*
primes-M-def prime-sum-upto-def)
also have $\dots = (\sum_{a \in n}. \sum_{a \in p} \mid \text{prime } p \wedge p \leq n. f \ n \ p)$
unfolding *f-def* **by** (*subst infsetsum-finite*) *auto*
also have $\dots = (\sum_{a \in n}. p) \in (SIGMA \ n:UNIV. \{p. \text{prime } p \wedge p \leq n\}). f \ n \ p)$
using *summable1* **by** (*subst infsetsum-Sigma*) *auto*
also have $\dots = (\sum_{a \in n}. p, n) \in (\lambda(n,p). (p, n)) \text{ ' } (SIGMA \ n:UNIV. \{p. \text{prime } p$
 $\wedge p \leq n\}). f \ n \ p)$
by (*subst infsetsum-reindex*) (*auto simp: case-prod-unfold inj-on-def*)
also have $(\lambda(n,p). (p, n)) \text{ ' } (SIGMA \ n:UNIV. \{p. \text{prime } p \wedge p \leq n\}) =$
 $(SIGMA \ p:\{p. \text{prime } p\}. \{p.. \}) \text{ by } auto$
also have $(\sum_{a \in n}. p, n) \in \dots f \ n \ p = (\sum_{a \in p} \mid \text{prime } p. \sum_{a \in \{p.. \}} f \ n \ p)$
using *summable2* **by** (*subst infsetsum-Sigma*) *auto*
also have $(\sum_{a \in p} \mid \text{prime } p. \sum_{a \in \{p.. \}} f \ n \ p) =$
 $(\sum_{a \in p} \mid \text{prime } p. \ln (\text{real } p) / \text{real } p * \text{hurwitz-zeta } p \ s)$
by (*intro infsetsum-cong eq*) *auto*
finally show *eval-fds fds-newman s* =
 $(\sum_{a \in p} \mid \text{prime } p. (\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \ s) .$

qed

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

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

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

where $\bar{f}(s)$ (which we shall call *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. $\zeta'(s)/\zeta(s)$ is meromorphic except for a single pole at $s = 1$ and one k -th order pole for any k -th order zero of ζ , but for the Prime Number Theorem, we are only concerned with the area $\Re(s) \geq 1$, where ζ 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 $A \equiv (\lambda p \ s. (s - 1) * \text{pre-zeta} \ (\text{real } p) \ s - \text{of-nat } p / (\text{of-nat } p \ \text{powr } s * (\text{of-nat } p \ \text{powr } s - 1)))$

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

begin

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

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

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 $\Re(s) > \frac{1}{2}$ 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* $s \ r$)

$(\lambda n \ s. \sum p < n. \text{if prime } p \ \text{then } B \ p \ s \ \text{else } 0)$ *pre-newman at-top*

proof –

from r **have** *Re*: $\text{Re } z > 1 / 2$ **if** *dist* $s \ z \leq r$ **for** z

using *abs-Re-le-cmod*[*of* $s - z$] r *that*

by (*auto simp*: *dist-norm abs-if split*: *if-splits*)

define x **where** $x = \text{Re } s - r$ — The lower bound for the real part in the disc

from r *Re* **have** $x > 1 / 2$ **by** (*auto simp*: *x-def*)

— The following sequence M bounds the summand, and it is obviously $O(n^{-1-\epsilon})$ and therefore summable

define C **where** $C = (\text{norm } s + r + 1) * (\text{norm } s + r) / x$

define M **where** $M = (\lambda p :: \text{nat}. \ln p * (C / p \ \text{powr } (x + 1) + 1 / (p \ \text{powr } x * (p \ \text{powr } x - 1))))$

show *?thesis unfolding pre-newman-def*

```

proof (intro Weierstrass-m-test-ev[OF eventually-mono[OF eventually-gt-at-top[of
1]]] ballI)
  show summable M
proof (rule summable-comparison-test-bigo)
  define  $\varepsilon$  where  $\varepsilon = \min (2 * x - 1) x / 2$ 
  from  $\langle x > 1 / 2 \rangle$  have  $\varepsilon > 0$   $1 + \varepsilon < 2 * x$   $1 + \varepsilon < x + 1$ 
  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-asymp)+
  from  $\varepsilon$  show summable  $(\lambda n. \text{norm } (n \text{ powr } (-1 - \varepsilon)))$ 
  by (simp add: summable-real-powr-iff)
qed
next
fix  $p :: \text{nat}$  and  $z$  assume  $p > 1$  and  $z: z \in \text{cball } s \ r$ 
from  $z \ r \ \text{Re}[of \ z]$  have  $x: \text{Re } z \geq x$   $x > 1 / 2$  and  $\text{Re } z > 1 / 2$ 
  using abs-Re-le-cmod[of  $s - z$ ] by (auto simp: x-def algebra-simps dist-norm)
have norm-z:  $\text{norm } z \leq \text{norm } s + r$ 
using  $z \ \text{norm-triangle-ineq2}[of \ z \ s] \ r$  by (auto simp: dist-norm norm-minus-commute)
from  $\langle p > 1 \rangle$  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:  $\text{norm } ((z - 1) * \text{pre-zeta } p \ z) \leq$ 
   $\text{norm } (z - 1) * (\text{norm } z / (\text{Re } z * p \ \text{powr } \text{Re } z))$ 
  using pre-zeta-bound'[of  $z \ p$ ]  $p \ \langle \text{Re } z > 1 / 2 \rangle$ 
  unfolding norm-mult by (intro mult-mono pre-zeta-bound) auto

have  $\text{norm } (B \ p \ z) = \ln \ p / p * \text{norm } (A \ p \ z)$ 
  unfolding B-def using  $\langle p > 1 \rangle$  by (simp add: B-def norm-mult norm-divide)
also have  $\dots \leq \ln \ p / p * (\text{norm } (z - 1) * \text{norm } z / \text{Re } z / p \ \text{powr } \text{Re } z +$ 
   $p / (p \ \text{powr } \text{Re } z * (p \ \text{powr } \text{Re } z - 1)))$ 
  unfolding A-def using  $\langle p > 1 \rangle$  and  $\langle \text{Re } z > 1 / 2 \rangle$  and bound
by (intro mult-left-mono order.trans[OF norm-triangle-ineq4 add-mono] mult-left-mono)
  (auto simp: norm-divide norm-mult norm-powr-real-powr
intro!: divide-left-mono order.trans[OF - norm-triangle-ineq2])
also have  $\dots = \ln \ p * (\text{norm } (z - 1) * \text{norm } z / \text{Re } z / p \ \text{powr } (\text{Re } z + 1)$ 
+
   $1 / (p \ \text{powr } \text{Re } z * (p \ \text{powr } \text{Re } z - 1)))$ 
  using  $\langle p > 1 \rangle$  by (simp add: field-simps powr-add powr-minus)
also have  $\text{norm } (z - 1) * \text{norm } z / \text{Re } z / p \ \text{powr } (\text{Re } z + 1) \leq C / p \ \text{powr}$ 
 $(x + 1)$ 
  unfolding C-def using  $r \ \langle \text{Re } z > 1 / 2 \rangle \ \text{norm-z } p \ x$ 
  by (intro mult-mono frac-le powr-mono order.trans[OF norm-triangle-ineq4])
auto
also have  $1 / (p \ \text{powr } \text{Re } z * (p \ \text{powr } \text{Re } z - 1)) \leq$ 
   $1 / (p \ \text{powr } x * (p \ \text{powr } x - 1))$  using  $\langle p > 1 \rangle \ x$ 
by (intro divide-left-mono mult-mono powr-mono diff-right-mono mult-pos-pos)
  (auto simp: ge-one-powr-ge-zero)
finally have  $\text{norm } (B \ p \ z) \leq M \ p$ 

```

```

    using ⟨p > 1⟩ by (simp add: mult-left-mono M-def)
    with ⟨M p ≥ 0⟩ show norm (if prime p then B p z else 0) ≤ M p by simp
  qed
qed

lemma sums-pre-newman: Re s > 1 / 2 ⇒ (λp. if prime p then B p s else 0)
sums pre-newman s
  using tendsto-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. Re s > 1 / 2}
proof -
  have holo: (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: 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-Re-le-cmod[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!: always-eventually holomorphic-on-imp-continuous-on
holomorphic-intros holo)
  qed
show ?thesis unfolding analytic-on-def
proof safe
  fix s assume Re s > 1 / 2
  thus ∃ r > 0. pre-newman holomorphic-on ball s r
  by (intro exI[of - (Re s - 1 / 2) / 2] conjI holo') (auto simp: field-simps)
qed
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 =

```

$\ln (\text{real } p) / \text{real } p * (p \text{ powr } (1 - s) / (s - 1) + \text{pre-zeta } p \ s)$
using s **by** (*auto simp add: hurwitz-zeta-def*)
also have $\dots = 1 / (s - 1) * (\ln (\text{real } p) / (p \text{ powr } s - 1) + B \ p \ s)$
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 $(\lambda p. (\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \ s)$ *abs-summable-on* $\{p. \text{prime } p\}$
using s **by** (*intro eval-fds-newman-conv-infsetsum*)
hence $(\lambda p. 1 / (s - 1) * (\ln (\text{real } p) / (p \text{ powr } s - 1) + B \ p \ s))$
abs-summable-on $\{p. \text{prime } p\}$
by (*subst (asm) abs-summable-on-cong[OF eq refl] auto*)
hence *summable*:
 $(\lambda p. \ln (\text{real } p) / (p \text{ powr } s - 1) + B \ p \ s)$ *abs-summable-on* $\{p. \text{prime } p\}$
using s **by** (*subst (asm) abs-summable-on-cmult-right-iff auto*)

from s **have** [*simp*]: $s \neq 1$ **by** *auto*
have *eval-fds fds-newman* $s =$
 $(\sum_{a \mid p} \text{prime } p. (\ln (\text{real } p) / \text{real } p) * \text{hurwitz-zeta } p \ s)$
using s **by** (*rule eval-fds-newman-conv-infsetsum*)
also have $\dots = (\sum_{a \mid p} \text{prime } p. 1 / (s - 1) * (\ln (\text{real } p) / (p \text{ powr } s - 1) + B \ p \ s))$
by (*intro infsetsum-cong eq auto*)
also have $\dots = 1 / (s - 1) * (\sum_{a \mid p} \text{prime } p. \ln (\text{real } p) / (p \text{ powr } s - 1) + B \ p \ s)$
(is - = * ?S) **by** (*rule infsetsum-cmult-right[OF summable]*)
also have $?S = (\sum p. \text{if prime } p \text{ then } \ln (\text{real } p) / (p \text{ powr } s - 1) + B \ p \ s \text{ else } 0)$
by (*subst infsetsum-nat[OF summable] auto*)
also have $\dots = (\sum p. (\text{if prime } p \text{ then } \ln (\text{real } p) / (p \text{ powr } s - 1) \text{ else } 0) + (\text{if prime } p \text{ then } B \ p \ s \text{ else } 0))$
by (*intro suminf-cong auto*)
also have $\dots = \text{pre-newman } s - \text{deriv zeta } s / \text{zeta } s$
using *sums-pre-newman[of s] sums-logderiv-zeta[of s] s*
by (*subst suminf-add [symmetric] (auto simp: sums-iff)*)
finally show *?thesis* **by** (*simp add: newman-def*)
qed

end

Next, we shall attempt to get rid of the pole by subtracting suitable multiples of $\zeta(s)$ and $\zeta'(s)$. To this end, we shall first prove the following alternative definition of $\zeta'(s)$:

lemma *deriv-zeta-eq'*:

assumes $0 < \text{Re } s \neq 1$

shows $\text{deriv zeta } s = \text{deriv } (\lambda z. \text{pre-zeta } 1 \ z * (z - 1)) \ s / (s - 1) - (\text{pre-zeta } 1 \ s * (s - 1) + 1) / (s - 1)^2$

```

    (is - = ?rhs)
  proof (rule DERIV-imp-deriv)
    have [derivative-intros]: (pre-zeta 1 has-field-derivative deriv (pre-zeta 1) s) (at
    s)
      by (intro holomorphic-derivI[of - UNIV] holomorphic-intros) auto
    have *: deriv ( $\lambda z. \text{pre-zeta } 1 z * (z - 1)$ ) s = deriv (pre-zeta 1) s * (s - 1) +
    pre-zeta 1 s
      by (subst deriv-mult)
      (auto intro!: holomorphic-on-imp-differentiable-at[of - UNIV] holomorphic-intros)
    hence (( $\lambda s. \text{pre-zeta } 1 s + 1 / (s - 1)$ ) has-field-derivative
      deriv (pre-zeta 1) s - 1 / ((s - 1) * (s - 1))) (at s)
      using assms by (auto intro!: derivative-eq-intros)
    also have deriv (pre-zeta 1) s - 1 / ((s - 1) * (s - 1)) = ?rhs
      using * assms by (simp add: divide-simps power2-eq-square, simp add: field-simps)
    also have (( $\lambda s. \text{pre-zeta } 1 s + 1 / (s - 1)$ ) has-field-derivative ?rhs) (at s)  $\longleftrightarrow$ 
      (zeta has-field-derivative ?rhs) (at s)
      using assms
      by (intro has-field-derivative-cong-ev eventually-mono[OF t1-space-nhds[of - 1]])
      (auto simp: zeta-def hurwitz-zeta-def)
    finally show ... .
  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 $\{s. \text{Re } s \geq 1\}$

proof

have *neq*: $\text{pre-zeta } 1 z * (z - 1) + 1 \neq 0$ **if** $\text{Re } z \geq 1$ **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 $?g = \lambda z. (1 - \text{inverse } (\text{pre-zeta } 1 z * (z - 1) + 1)) * ((z - 1) * \text{deriv } ((\lambda u. \text{pre-zeta } 1 u * (u - 1))) z - (\text{pre-zeta } 1 z * (z - 1) + 1))$

show ($\lambda z. \text{if } z = 1 \text{ then } \text{deriv } ?g 1 \text{ else } (z - 1) * \text{deriv zeta } z - \text{deriv zeta } z / \text{zeta } z$)

analytic-on $\{s. \text{Re } s \geq 1\}$ (**is** $?f$ *analytic-on* -)

proof (*rule pole-theorem-analytic-0*)

show $?g$ *analytic-on* $\{s. 1 \leq \text{Re } s\}$ **using** *neq*

by (*auto intro!: analytic-intros*)

next

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

if $z \in \{s. 1 \leq \text{Re } s\}$ **for** z

proof -

have *: *isCont* ($\lambda z. \text{pre-zeta } 1 z * (z - 1) + 1$) z

by (*auto intro!: continuous-intros*)

obtain *e* **where** $e > 0$ **and** $e: \bigwedge y. \text{dist } z y < e \implies \text{pre-zeta } (\text{Suc } 0) y * (y - 1) + 1 \neq 0$

using *continuous-at-avoid* [OF * *neq*[of z]] z **by** *auto*

show *thesis*

```

proof (intro exI ballI conjI)
  fix w
  assume w: w ∈ ball z (min e 1) - {1}
  then have Re w > 0
    using complex-Re-le-cmod [of z-w] z by (simp add: dist-norm)
  with w show ?g w = (w - 1) * (if w = 1 then deriv ?g 1 else
    (w - 1) * deriv zeta w - deriv zeta w / zeta w)
    by (subst (1 2) deriv-zeta-eq',
      simp-all add: zeta-def hurwitz-zeta-def divide-simps e power2-eq-square)
      (simp-all add: algebra-simps)?
  qed (use <e > 0) in auto)
qed
qed auto
qed

```

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

lemma *analytic-newman-variant*:

```

obtains c a where
  (λz. if z = 1 then a else newman z + deriv zeta z + c * zeta z) analytic-on
  {s. Re s ≥ 1}
proof -
  obtain c where
    c: (λz. if z = 1 then c else (z - 1) * deriv zeta z - deriv zeta z / zeta z)
      analytic-on {s. Re s ≥ 1}
  using analytic-zeta-derivdiff by blast
  let ?g = λz. pre-newman z +
    (if z = 1 then c
      else (z - 1) * deriv zeta z -
        deriv zeta z / zeta z) - (c + pre-newman 1) * (pre-zeta 1 z * (z -
  1) + 1)
  have (λz. if z = 1 then deriv ?g 1 else newman z + deriv zeta z + -(c +
  pre-newman 1)) * zeta z
    analytic-on {s. Re s ≥ 1} (is ?f analytic-on -)
  proof (rule pole-theorem-analytic-0)
  show ?g analytic-on {s. 1 ≤ Re s}
    by (intro c analytic-intros) auto
  next
  show ∃ d>0. ∀ w∈ball z d - {1}. ?g w = (w - 1) * ?f w
    if z ∈ {s. 1 ≤ Re s} for z using that
    by (intro exI[of - 1], simp-all add: newman-def divide-simps zeta-def hur-
  witz-zeta-def)
    (auto simp: field-simps)?
  qed auto
  with that show ?thesis by blast
qed

```

4.2 The asymptotic expansion of \mathfrak{M}

Our next goal is to show the key result that $\mathfrak{M}(x) = \ln n + c + o(1)$.

As a first step, we invoke Ingham's Tauberian theorem on the function we have just defined and obtain that the sum

$$\sum_{n=1}^{\infty} \frac{\mathfrak{M}(n) - \ln n + c}{n}$$

exists.

lemma *mertens-summable*:

obtains $c :: \text{real}$ **where** *summable* $(\lambda n. (\mathfrak{M} n - \ln n + c) / n)$

proof –

from *analytic-newman-variant* **obtain** c **a where**

analytic: $(\lambda z. \text{if } z = 1 \text{ then } a \text{ else newman } z + \text{deriv } \zeta z + c * \zeta z)$
analytic-on $\{s. \text{Re } s \geq 1\}$.

define f **where** $f = (\lambda z. \text{if } z = 1 \text{ then } a \text{ else newman } z + \text{deriv } \zeta z + c * \zeta z)$

have *analytic*: f *analytic-on* $\{s. \text{Re } s \geq 1\}$ **using** *analytic* **by** (*simp add: f-def*)

define F **where** $F = \text{fds-newman} + \text{fds-deriv } \text{fds-zeta} + \text{fds-const } c * \text{fds-zeta}$

note $le = \text{conv-abscissa-add-leI conv-abscissa-deriv-le conv-abscissa-newman conv-abscissa-mult-const-left}$

note *intros* = le $le[\text{THEN } le\text{-less-trans}] le[\text{THEN } \text{order.trans}] \text{fds-converges}$

have *eval-F*: $\text{eval-fds } F s = f s$ **if** $s: \text{Re } s > 1$ **for** s

proof –

have *eval-fds* $F s = \text{eval-fds } (\text{fds-newman} + \text{fds-deriv } \text{fds-zeta}) s +$
 $\text{eval-fds } (\text{fds-const } c * \text{fds-zeta}) s$

unfolding $F\text{-def}$ **using** s **by** (*subst eval-fds-add*) (*auto intro!: intros*)

also have $\dots = f s$ **using** s **unfolding** $f\text{-def}$

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*: $\text{fds-converges } F s$ **if** $\text{Re } s \geq 1$ **for** s

proof (*rule Newman-Ingham-1*)

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

using *mertens-bounded* **by** (*rule landau-o.big.compose*) *real-asymp*

from *natfun-bigo-1E[OF this, of 1]*

obtain c' **where** $c': c' \geq 1 \wedge \lambda n. |\mathfrak{M} (\text{real } n) - \ln (\text{real } n)| \leq c'$ **by** *auto*

have *Bseq* $(\text{fds-nth } F)$

proof (*intro BseqI allI*)

fix $n :: \text{nat}$

show $\text{norm } (\text{fds-nth } F n) \leq (c' + \text{norm } c)$ **unfolding** $F\text{-def}$ **using** c'

by (*auto simp: fds-nth-zeta fds-nth-deriv fds-nth-newman scaleR-conv-of-real in-Reals-norm*)

intro!: order.trans[OF norm-triangle-ineq] add-mono)

qed (*insert c', auto intro: add-pos-nonneg*)

thus $\text{fds-nth } F \in O(\lambda-. 1)$ **by** (*simp add: natfun-bigo-iff-Bseq*)

next


```

  show  $f$  analytic-on  $\{s. \operatorname{Re} s \geq 1\}$  by fact
next
  show eval-fds  $F s = f s$  if  $\operatorname{Re} s > 1$  for  $s$  using that by (rule eval-F)
qed (insert that, auto simp: F-def intro!: intros)
from conv[of 1] have summable  $(\lambda n. \operatorname{fds}\text{-nth } F n / \operatorname{of}\text{-nat } n)$ 
  unfolding fds-converges-def by auto
also have ?this  $\longleftrightarrow$  summable  $(\lambda n. (\mathfrak{M} n - \operatorname{Ln} n + c) / n)$ 
  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  $(\lambda n. (\mathfrak{M} n - \operatorname{Re} (\operatorname{of}\text{-nat } n)) + \operatorname{Re} c) / n)$ 
  by (auto dest: summable-Re)
also have ?this  $\longleftrightarrow$  summable  $(\lambda n. (\mathfrak{M} n - \operatorname{ln} n + \operatorname{Re} c) / n)$ 
  by (intro summable-cong eventually-mono[OF eventually-gt-at-top[of 0]]) (auto
intro!: sum.cong)
  finally show ?thesis using that[of Re c] by blast
qed

```

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  $d::\operatorname{real}$ 
  assumes  $d: |\sum i = M..N. a i / i| < d$  and  $le: \bigwedge n. a n + \operatorname{ln} n \leq a (\operatorname{Suc} n) + \operatorname{ln} (\operatorname{Suc} n)$ 
    and  $0 < M < N$ 
  shows  $a M \leq d * N / (\operatorname{real} N - \operatorname{real} M) + (\operatorname{real} N - \operatorname{real} M) / M \wedge$ 
     $-a N \leq d * N / (\operatorname{real} N - \operatorname{real} M) + (\operatorname{real} N - \operatorname{real} M) / M$ 
proof -
  have  $0 \leq d$ 
  using assms by linarith+
  then have  $0 \leq d * N / (N - M + 1)$  by simp
  then have  $le\text{-dN}: [0 \leq x \implies x \leq d * N / (N - M + 1)] \implies x \leq d * N / (N - M + 1)$  for  $x::\operatorname{real}$ 
  by linarith
  have  $le\text{-a-ln}: a m + \operatorname{ln} m \leq a n + \operatorname{ln} n$  if  $n \geq m$  for  $n m$ 
  by (rule transitive-stepwise-le) (use le that in auto)
  have  $*$ :  $x \leq b \wedge y \leq b$  if  $a \leq b \wedge x \leq a \wedge y \leq a$  for  $a b x y::\operatorname{real}$ 
  using that by linarith
  show ?thesis
  proof (rule *)
    show  $d * N / (N - M) + \operatorname{ln} (N / M) \leq d * N / (\operatorname{real} N - \operatorname{real} M) + (\operatorname{real} N - \operatorname{real} M) / M$ 
    using  $\langle 0 < M \rangle \langle M < N \rangle \operatorname{ln}\text{-le}\text{-minus}\text{-one} [of N / M]$ 
    by (simp add: of-nat-diff) (simp add: divide-simps)
  next
    have  $a M - \operatorname{ln} (N / M) \leq (d * N) / (N - M + 1)$ 
    proof (rule le-dN)
      assume  $0: 0 \leq a M - \operatorname{ln} (N / M)$ 

```

```

have  $(\text{Suc } N - M) * (a M - \ln (N / M)) / N = (\sum i = M..N. (a M - \ln (N / M)) / N)$ 
by simp
also have  $\dots \leq (\sum i = M..N. a i / i)$ 
proof (rule sum-mono)
  fix  $i$ 
  assume  $i: i \in \{M..N\}$ 
  with  $\langle 0 < M \rangle$  have  $0 < i$  by auto
  have  $(a M - \ln (N / M)) / N \leq (a M - \ln (N / M)) / i$ 
using  $0$  using  $i \langle 0 < M \rangle$  by (simp add: frac-le-eq divide-simps mult-left-mono)
also have  $a M + \ln (\text{real } M) \leq a i + \ln (\text{real } N)$ 
  by (rule order.trans[OF le-a-ln[of M i]]) (use i assms in auto)
hence  $(a M - \ln (N / M)) / i \leq a i / \text{real } i$ 
  using assms i by (intro divide-right-mono) (auto simp: ln-div field-simps)
finally show  $(a M - \ln (N / M)) / \text{real } N \leq a i / \text{real } i$  .
qed
finally have  $((\text{Suc } N) - M) * (a M - \ln (N / M)) / N \leq |\sum i = M..N. a i / i|$ 
by simp
also have  $\dots \leq d$  using  $d$  by simp
finally have  $((\text{Suc } N) - M) * (a M - \ln (N / M)) / N \leq d$  .
then show ?thesis
  using  $\langle M < N \rangle$  by (simp add: of-nat-diff field-simps)
qed
also have  $\dots \leq d * N / (N - M)$ 
  using assms(1,4) by (simp add: field-simps)
finally show  $a M \leq d * N / (N - M) + \ln (N / M)$  by simp
next
have  $- a N - \ln (N / M) \leq (d * N) / (N - M + 1)$ 
proof (rule le-dN)
  assume  $0: 0 \leq - a N - \ln (N / M)$ 
  have  $(\sum i = M..N. a i / i) \leq (\sum i = M..N. (a N + \ln (N / M)) / N)$ 
proof (rule sum-mono)
  fix  $i$ 
  assume  $i: i \in \{M..N\}$ 
  with  $\langle 0 < M \rangle$  have  $0 < i$  by auto
  have  $a i + \ln (\text{real } M) \leq a N + \ln (\text{real } N)$ 
  by (rule order.trans[OF - le-a-ln[of i N]]) (use i assms in auto)
hence  $a i / i \leq (a N + \ln (N / M)) / i$ 
  using assms(3,4) by (intro divide-right-mono) (auto simp: field-simps ln-div)
also have  $\dots \leq (a N + \ln (N / M)) / N$ 
  using  $i \langle i > 0 \rangle 0$  by (intro divide-left-mono-neg) auto
finally show  $a i / i \leq (a N + \ln (N / M)) / N$  .
qed
also have  $\dots = ((\text{Suc } N) - M) * (a N + \ln (N / M)) / N$ 
  by simp
finally have  $(\sum i = M..N. a i / i) \leq (\text{real } (\text{Suc } N) - \text{real } M) * (a N + \ln (N / M)) / N$ 

```

```

    using ⟨M < N⟩ by (simp add: of-nat-diff)
    then have -((real (Suc N) - real M) * (a N + ln (N / M)) / N) ≤ |∑ i
= M..N. a i / i|
      by linarith
    also have ... ≤ d using d by simp
    finally have -((real (Suc N) - real M) * (a N + ln (N / M)) / N) ≤ d .
    then show ?thesis
      using ⟨M < N⟩ by (simp add: of-nat-diff field-simps)
  qed
  also have ... ≤ d * N / real (N - M)
    using ⟨0 < M⟩ ⟨M < N⟩ ⟨0 ≤ d⟩ by (simp add: field-simps)
  finally show -a N ≤ d * N / real (N - M) + ln (N / M) by simp
  qed
  qed

```

proposition *sum-goestozero-theorem:*

```

  assumes summ: summable (λi. a i / i)
    and le: ∧n. a n + ln n ≤ a (Suc n) + ln (Suc n)
    shows a ⟶ 0
  proof (clarisimp simp: lim-sequentially)
    fix r::real
    assume r > 0
    have *: ∃n0. ∀n≥n0. |a n| < ε if ε: 0 < ε ε < 1 for ε
    proof -
      have 0 < (ε / 8)2 using ⟨0 < ε⟩ by simp
      then obtain N0 where N0: ∧m n. m ≥ N0 ⟹ norm (∑ k=m..n. (λi. a i
/ i) k) < (ε / 8)2
        by (metis summable-partial-sum-bound summ)
      obtain N1 where real N1 > 4 / ε
        using reals-Archimedean2[of 4 / ε] ε by auto
      hence N1 ≠ 0 and N1: 1 / real N1 < ε / 4 using ε
        by (auto simp: divide-simps mult-ac intro: Nat.gr0I)

      have |a n| < ε if n: n ≥ 2 * N0 + N1 + 7 for n
      proof -
        define k where k = ⌊n * ε / 4⌋
        have n * ε / 4 > 1 and n * ε / 4 ≤ n / 4 and n / 4 < n
          using less-le-trans[OF N1, of n / N1 * ε / 4] ⟨N1 ≠ 0⟩ ε n by (auto simp:
field-simps)
        hence k: k > 0 4 * k ≤ n nat k < n (n * ε / 4) - 1 < k k ≤ (n * ε / 4)
          unfolding k-def by linarith+

        have -a n < ε
        proof -
          have N0 ≤ n - nat k
            using n k by linarith
          then have *: |∑ k = n - nat k .. n. a k / k| < (ε / 8)2
            using N0 [of n - nat k n] by simp
          have -a n ≤ (ε / 8)2 * n / ⌊n * ε / 4⌋ + ⌊n * ε / 4⌋ / (n - k)

```

```

    using sum-goestozero-lemma [OF * le, THEN conjunct2] k by (simp add:
of-nat-diff k-def)
  also have ... < ε
  proof -
    have ε / 16 * n / k < 2
      using k by (auto simp: field-simps)
    then have ε * (ε / 16 * n / k) < ε * 2
      using ε mult-less-cancel-left-pos by blast
    then have (ε / 8)2 * n / k < ε / 2
      by (simp add: field-simps power2-eq-square)
    moreover have k / (n - k) < ε / 2
    proof -
      have (ε + 2) * k < 4 * k using k ε by simp
      also have ... ≤ ε * 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 n < ε
proof -
  have N0 ≤ n using n k by linarith
  then have *: |∑ k = n .. n + nat k. a k / k| < (ε/8)2
    using N0 [of n n + nat k] by simp
  have a n ≤ (ε/8)2 * (n + nat k) / k + k / n
    using sum-goestozero-lemma [OF * le, THEN conjunct1] k by (simp add:
of-nat-diff)
  also have ... < ε
  proof -
    have 4 ≤ 28 * real-of-int k using k by linarith
    then have ε/16 * n / k < 2 using k by (auto simp: field-simps)
    have ε * (real n + k) < 32 * k
    proof -
      have ε * n / 4 < k + 1 by (simp add: mult.commute k-def)
      then have ε * n < 4 * k + 4 by (simp add: divide-simps)
      also have ... ≤ 8 * k using k by auto
      finally have 1: ε * real n < 8 * k .
      have 2: ε * k < k using k ε by simp
      show ?thesis using k add-strict-mono [OF 1 2] by (simp add: alge-
bra-simps)
    qed
    then have (ε / 8)2 * real (n + nat k) / k < ε / 2
      using ε k by (simp add: divide-simps mult-less-0-iff power2-eq-square)
    moreover have k / n < ε / 2
      using k ε by (auto simp: k-def field-simps)
    ultimately show ?thesis by linarith
  qed
  finally show ?thesis .

```

qed
ultimately show *?thesis by force*
qed
then show *?thesis by blast*
qed
show $\exists n0. \forall n \geq n0. |a\ n| < r$
using * [of min r (1/5)] $\langle 0 < r \rangle$ **by force**
qed

This leads us to the main intermediate result:

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

corollary *M-minus-ln-limit:*

obtains *c where ((λx::real. M x - ln x) → c) at-top*
proof –
from *Mertens-convergent* **obtain** *c where (λn. M n - ln n) → c*
by *(auto simp: convergent-def)*
hence 1: $((\lambda x::real. \mathfrak{M}\ (\text{nat } \lfloor x \rfloor) - \ln\ (\text{nat } \lfloor x \rfloor)) \longrightarrow c)$ *at-top*
by *(rule filterlim-compose) real-asymp*
have 2: $((\lambda x::real. \ln\ (\text{nat } \lfloor x \rfloor) - \ln\ x) \longrightarrow 0)$ *at-top*
by *real-asymp*
have 3: $((\lambda x. \mathfrak{M}\ x - \ln\ x) \longrightarrow c)$ *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:

$$\begin{aligned} \vartheta(x) &= x \ln x + cx + o(x) - \int_2^x \ln t + c + o(1) dt \\ &= x \ln x + cx + o(x) - (x \ln x - x + cx + o(x)) \\ &= x + o(x) \end{aligned}$$

In conclusion, $\vartheta(x) \sim x$.

theorem ϑ -asymptotics: $\vartheta \sim[at-top] (\lambda x. x)$

proof –

from \mathfrak{M} -minus-ln-limit **obtain** c **where** $c: ((\lambda x. \mathfrak{M} x - \ln x) \longrightarrow c)$ *at-top*
by *auto*

define r **where** $r = (\lambda x. \mathfrak{M} x - \ln x - c)$

have \mathfrak{M} -expand: $\mathfrak{M} = (\lambda x. \ln x + c + r x)$

by (*simp add: r-def*)

have $r: r \in o(\lambda \cdot 1)$ **unfolding** *r-def*

using *tendsto-add[OF c tendsto-const[of -c]]* **by** (*intro smalloI-tendsto*) *auto*

define r' **where** $r' = (\lambda x. \text{integral } \{2..x\} r)$

have *integrable-r: r integrable-on {x..y}*

if $2 \leq x$ **for** $x y :: \text{real}$ **using** *that* **unfolding** *r-def*

by (*intro integrable-diff integrable-primes-M*)

(*auto intro!: integrable-continuous-real continuous-intros*)

hence *integral: (r has-integral r' x) {2..x}* **if** $x \geq 2$ **for** x

by (*auto simp: has-integral-iff r'-def*)

have $r': r' \in o(\lambda x. x)$ **using** *integrable-r* **unfolding** *r'-def*

by (*intro integral-smallo[OF r]*) (*auto simp: filterlim-ident*)

define C **where** $C = 2 * (c + \ln 2 - 1)$

have $\vartheta \sim[at-top] (\lambda x. x + (r x * x + C - r' x))$

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

fix $x :: \text{real}$ **assume** $x > 2$

have (\mathfrak{M} *has-integral* $((x * \ln x - x + c * x) - (2 * \ln 2 - 2 + c * 2) + r' x)$) $\{2..x\}$

unfolding \mathfrak{M} -expand **using** x

by (*intro has-integral-add[OF fundamental-theorem-of-calculus integral]*)

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

intro!: derivative-eq-intros continuous-intros)

from *has-integral-unique[OF ϑ -conv- \mathfrak{M} -integral this]*

show $\vartheta x = x + (r x * x + C - r' x)$ **using** x

by (*simp add: field-simps \mathfrak{M} -expand C-def*)

qed

also have $(\lambda x. r x * x + C - r' x) \in o(\lambda x. x)$

proof (*intro sum-in-smallo r*)

show $(\lambda-. C) \in o(\lambda x. x)$ **by** *real-asymp*
qed (*insert landau-o.small-big-mult[OF r, of $\lambda x. x$] r', simp-all*)
hence $(\lambda x. x + (r x * x + C - r' x)) \sim[at-top] (\lambda x. x)$
by (*subst asymp-equiv-add-right*) *auto*
finally show *?thesis* **by** *auto*
qed

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

corollary *ψ -asymptotics*: $\psi \sim[at-top] (\lambda x. x)$
using *ϑ -asymptotics PNT4-imp-PNT5* **by** *simp*

corollary *prime-number-theorem*: $\pi \sim[at-top] (\lambda x. x / \ln x)$
using *ϑ -asymptotics PNT4-imp-PNT1* **by** *simp*

corollary *ln- π -asymptotics*: $(\lambda x. \ln (\pi x)) \sim[at-top] \ln$
using *prime-number-theorem PNT1-imp-PNT1'* **by** *simp*

corollary *π -ln- π -asymptotics*: $(\lambda x. \pi x * \ln (\pi x)) \sim[at-top] (\lambda x. x)$
using *prime-number-theorem PNT1-imp-PNT2* **by** *simp*

corollary *nth-prime-asymptotics*: $(\lambda n. \text{real } (nth\text{-prime } n)) \sim[at-top] (\lambda n. \text{real } n * \ln (\text{real } n))$
using *π -ln- π -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''*:
 $(\lambda x. \text{card } \{p. \text{prime } p \wedge \text{real } p \leq x\}) \sim[at-top] (\lambda x. x / \ln x)$

proof –
have $\pi = (\lambda x. \text{card } \{p. \text{prime } p \wedge \text{real } p \leq x\})$
by (*intro ext*) (*simp add: π -def prime-sum-upto-def*)
with *prime-number-theorem* **show** *?thesis* **by** *simp*
qed

corollary *prime-number-theorem'''*:
 $(\lambda n. \text{card } \{p. \text{prime } p \wedge p \leq n\}) \sim[at-top] (\lambda n. \text{real } n / \ln (\text{real } n))$

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

end

5 Mertens' Theorems

```
theory Mertens-Theorems
imports
  Prime-Counting-Functions
  Stirling-Formula.Stirling-Formula
begin
```

In this section, we will prove Mertens' First and Second Theorem. These are weaker results than the Prime Number Theorem, and we will derive them without using it.

However, like Mertens himself, we will not only prove them *asymptotically*, but *absolutely*. This means that we will show that the remainder terms are not only “Big-O” of some bound, but we will give concrete (and reasonably tight) upper and lower bounds for them that hold on the entire domain. This makes the proofs a bit more tedious.

5.1 Absolute Bounds for Mertens' First Theorem

We have already shown the asymptotic form of Mertens' first theorem, i. e. $\mathfrak{M}(n) = \ln n + O(1)$. We now want to obtain some absolute bounds on the $O(1)$ remainder term using a more careful derivation than before.

The precise bounds we will show are $\mathfrak{M}(n) - \ln n \in (-1 - \frac{9}{\pi^2}; \ln 4] \approx (-1.9119; 1.3863]$ for $n \in \mathbb{N}$.

First, we need a simple lemma on the finiteness of exponents to consider in a sum of all prime powers up to a certain point:

```
lemma exponents-le-finite:
  assumes  $p > (1 :: nat)$   $k > 0$ 
  shows  $\text{finite } \{i. \text{real } (p \wedge (k * i + l)) \leq x\}$ 
proof (rule finite-subset)
  show  $\{i. \text{real } (p \wedge (k * i + l)) \leq x\} \subseteq \{.. \text{nat } \lfloor x \rfloor\}$ 
proof safe
  fix  $i$  assume  $i: \text{real } (p \wedge (k * i + l)) \leq x$ 
  have  $i < 2 \wedge i$  by (rule less-exp)
  also from  $\text{assms}$  have  $i \leq k * i + l$  by (cases  $k$ ) auto
  hence  $2 \wedge i \leq (2 \wedge (k * i + l)) :: \text{nat}$ 
  using  $\text{assms}$  by (intro power-increasing) auto
  also have  $\dots \leq p \wedge (k * i + l)$  using  $\text{assms}$  by (intro power-mono) auto
  also have  $\text{real } \dots \leq x$  using  $i$  by simp
  finally show  $i \leq \text{nat } \lfloor x \rfloor$  by linarith
qed
qed auto
```


Next, we need the following bound on $\zeta'(2)$:

lemma *deriv-zeta-2-bound*: $Re (deriv\ zeta\ 2) > -1$

proof –

have $((\lambda x::real. \ln (x + 3) * (x + 3) \text{ powr } -2) \text{ has-integral } (\ln 3 + 1) / 3)$
(interior {0..})

using *ln-powr-has-integral-at-top*[of 1 0 3 -2]

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

hence $((\lambda x::real. \ln (x + 3) * (x + 3) \text{ powr } -2) \text{ has-integral } (\ln 3 + 1) / 3)$
{0..}

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

also have $?this \longleftrightarrow ((\lambda x::real. \ln (x + 3) / (x + 3) ^ 2) \text{ has-integral } (\ln 3 + 1) / 3) \{0..\}$

by *(intro has-integral-cong) (auto simp: powr-minus field-simps)*

finally have *int:*

have $\exp (1 / 2 :: real) ^ 2 \leq 2 ^ 2$

using *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*: $\ln x / x ^ 2 \leq \ln y / y ^ 2$ **if** $y \geq \exp (1/2) x \geq y$ **for** $x y :: real$

proof *(rule DERIV-nonpos-imp-nonincreasing*[of - - $\lambda x. \ln x / x ^ 2$])

fix t **assume** $t: t \geq y \geq x$

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 $t \langle y > 0 \rangle$ **have** $((\lambda x. \ln x / x ^ 2) \text{ has-field-derivative } ((1 - 2 * \ln t) / t ^ 3))$ *(at t)*

by *(auto intro!: derivative-eq-intros simp: eval-nat-numeral field-simps)*

moreover have $(1 - 2 * \ln t) / t ^ 3 \leq 0$

using t **that** $\langle y > 0 \rangle \langle \ln t \geq 1 / 2 \rangle$ **by** *(intro divide-nonpos-pos) auto*

ultimately show $\exists f'. ((\lambda x. \ln x / x ^ 2) \text{ has-field-derivative } f')$ *(at t) $\wedge f' \leq$*

0 **by** *blast*

qed *fact+*

have *fds-converges* *(fds-deriv fds-zeta)* $(2 :: complex)$

by *(intro fds-converges-deriv) auto*

hence $(\lambda n. \text{ of-real } (-\ln (\text{real } (\text{Suc } n)) / (\text{of-nat } (\text{Suc } n)) ^ 2)) \text{ sums deriv zeta } 2$

by *(auto simp: fds-converges-altdef add-ac eval-fds-deriv-zeta fds-nth-deriv scaleR-conv-of-real*

simp del: of-nat-Suc)

note $*$ = *sums-split-initial-segment*[OF *sums-minus*[OF *sums-Re*[OF *this*]], of 3]

have $(\lambda n. \ln (\text{real } (n+4)) / \text{real } (n+4) ^ 2) \text{ sums } (-\text{Re } (\text{deriv zeta } 2) - (\ln 2 / 4 + \ln 3 / 9))$

using $*$ **by** *(simp add: eval-nat-numeral)*

hence $-\text{Re } (\text{deriv zeta } 2) - (\ln 2 / 4 + \ln 3 / 9) =$

$(\sum n. \ln (\text{real } (\text{Suc } n) + 3) / (\text{real } (\text{Suc } n) + 3) ^ 2)$

by *(simp-all add: sums-iff algebra-simps)*

also have $\dots \leq (\ln 3 + 1) / 3$ **using** *int exp-half*
by (*intro decreasing-sum-le-integral divide-nonneg-pos mono*) (*auto simp: powr-minus field-simps*)
finally have $-Re(\text{deriv zeta } 2) \leq (16 * \ln 3 + 9 * \ln 2 + 12) / 36$
by *simp*
also have $\ln 3 \leq (11 / 10 :: \text{real})$
using *ln-approx-bounds[of 3 2]* **by** (*simp add: power-numeral-reduce numeral-2-eq-2*)
hence $(16 * \ln 3 + 9 * \ln 2 + 12) / 36 \leq (16 * (11 / 10) + 9 * 25 / 36 + 12) / (36 :: \text{real})$
using *ln2-le-25-over-36* **by** (*intro add-mono mult-left-mono divide-right-mono*)
auto
also have $\dots < 1$ **by** *simp*
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 $R \equiv (\sum (p, i) \mid \text{prime } p \wedge i > 1 \wedge \text{real } (p \wedge i) \leq x. \ln(\text{real } p) / p \wedge i)$

shows $R < 9 / \pi^2$

proof –

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

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

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

proof (*rule finite-subset*)

show $\{i. i > 1 \wedge \text{real } (p \wedge (i + k)) \leq x\} \subseteq \{.. \text{nat } \lfloor x \rfloor\}$

proof *safe*

fix i **assume** $i: i > 1 \text{ real } (p \wedge (i + k)) \leq x$

have $i < 2 \wedge (i + k)$ **by** (*induction i*) *auto*

also from p **have** $\dots \leq p \wedge (i + k)$ **by** (*intro power-mono*) (*auto dest: prime-gt-1-nat*)

also have $\text{real } \dots \leq x$ **using** i **by** *simp*

finally show $i \leq \text{nat } \lfloor x \rfloor$ **by** *linarith*

qed

qed *auto*

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

proof *safe*

fix p i **assume** $pi: \text{prime } p \text{ real } (p \wedge \text{Suc } i) \leq x \ i > 1$

have $\text{real } (p \wedge i) \leq \text{real } (p \wedge \text{Suc } i)$

using pi **unfolding** *of-nat-le-iff* **by** (*intro power-increasing*) (*auto dest: prime-gt-1-nat*)

also have $\dots \leq x$ **by** *fact*

finally show $\text{real } (p \wedge i) \leq x$.
qed

have $S'\text{-alt}$: $S' = (\text{SIGMA } p:\{p. \text{prime } p \wedge \text{real } p \leq x\}. \{i. i > 1 \wedge \text{real } (p \wedge i) \leq x\})$
unfolding $S'\text{-def}$
proof safe
fix $p\ i$ **assume** $\text{prime } p \text{ real } (p \wedge i) \leq x\ i > 1$
hence $p \wedge 1 \leq p \wedge i$
by (*intro power-increasing*) (*auto dest: prime-gt-1-nat*)
also have $\text{real } \dots \leq x$ **by fact**
finally show $\text{real } p \leq x$ **by simp**
qed

have finite : $\text{finite } \{p. \text{prime } p \wedge \text{real } p \leq x\}$
by (*rule finite-subset[OF - finite-Nats-le-real[of x]]*) (*auto dest: prime-gt-0-nat*)
have $\text{finite } S'$ **unfolding** $S'\text{-alt}$ **using** $\text{finite-row}[of - 0]$
by (*intro finite-SigmaI finite*) *auto*

have $R \leq 3 / 2 * (\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge i))$
proof –
have $R = (\sum y \in \{0, 1\}. \sum z \mid z \in S' \wedge \text{snd } z \bmod 2 = y. \ln (\text{real } (\text{fst } z)) / \text{real } (\text{fst } z \wedge \text{snd } z))$
using $\langle \text{finite } S' \rangle$ **by** (*subst sum.group*) (*auto simp: case-prod-unfold R-def S'-def*)
also have $\dots = (\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge i)) + (\sum (p, i) \mid (p, i) \in S' \wedge \text{odd } i. \ln (\text{real } p) / \text{real } (p \wedge i))$
unfolding *even-iff-mod-2-eq-zero odd-iff-mod-2-eq-one* **by** (*simp add: case-prod-unfold*)
also have $(\sum (p, i) \mid (p, i) \in S' \wedge \text{odd } i. \ln (\text{real } p) / \text{real } (p \wedge i)) = (\sum (p, i) \mid (p, i) \in S'' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge \text{Suc } i))$
by (*intro sum.reindex-bij-witness[of - $\lambda(p, i). (p, \text{Suc } i)$ $\lambda(p, i). (p, i - 1)$]*) (*auto simp: case-prod-unfold S'-def S''-def elim: oddE simp del: power-Suc*)
also have $\dots \leq (\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge \text{Suc } i))$
using $\langle S'' \subseteq S' \rangle$ **unfolding** *case-prod-unfold*
by (*intro sum-mono2 divide-nonneg-pos ln-ge-zero finite-subset[OF - $\langle \text{finite } S' \rangle$]*)
(auto simp: S'-def S''-def case-prod-unfold dest: prime-gt-0-nat simp del: power-Suc)
also have $\dots \leq (\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (2 * p \wedge i))$
unfolding *case-prod-unfold*
by (*intro sum-mono divide-left-mono*) (*auto simp: S'-def dest!: prime-gt-1-nat*)
also have $\dots = (1 / 2) * (\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge i))$
by (*subst sum-distrib-left*) (*auto simp: case-prod-unfold*)
also have $(\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge i)) + \dots = 3 / 2 * (\sum (p, i) \mid (p, i) \in S' \wedge \text{even } i. \ln (\text{real } p) / \text{real } (p \wedge i))$
by simp
finally show *?thesis* **by simp**
qed

also have $(\sum (p,i) \mid (p,i) \in S' \wedge \text{even } i. \ln(\text{real } p) / \text{real}(p \wedge i)) =$
 $(\sum p \mid \text{prime } p \wedge \text{real } p \leq x. \ln(\text{real } p) * (\sum i \mid i > 0 \wedge \text{even } i \wedge \text{real}(p \wedge i) \leq x. (1 / \text{real } p) \wedge i))$
unfolding *sum-distrib-left*
proof (*subst sum.Sigma[OF - ballI]*)
fix p **assume** $p: p \in \{p. \text{prime } p \wedge \text{real } p \leq x\}$
thus $\text{finite } \{i. 0 < i \wedge \text{even } i \wedge \text{real}(p \wedge i) \leq x\}$
by (*intro finite-subset[OF - exponents-le-finite[of p 1 0 x]]*) (*auto dest: prime-gt-1-nat*)
qed (*auto intro!: sum.cong finite-subset[OF - finite-Nats-le-real[of x]]*)
dest: prime-gt-0-nat simp: S'-alt power-divide)
also have $\dots \leq (\sum p \mid \text{prime } p \wedge \text{real } p \leq x. \ln(\text{real } p) / (\text{real } p \wedge 2 - 1))$
proof (*rule sum-mono*)
fix p **assume** $p: p \in \{p. \text{prime } p \wedge \text{real } p \leq x\}$
have $p > 1$ **using** p **by** (*auto dest: prime-gt-1-nat*)
have $(\sum i \mid i > 0 \wedge \text{even } i \wedge \text{real}(p \wedge i) \leq x. (1 / \text{real } p) \wedge i) =$
 $(\sum i \mid \text{real}(p \wedge (2 * i + 2)) \leq x. (1 / \text{real } p) \wedge (2 * i)) / \text{real } p \wedge 2$
(is - = ?S / -) **unfolding** *sum-divide-distrib*
by (*rule sum.reindex-bij-witness[of - $\lambda i. 2 * \text{Suc } i \lambda i. (i - 2) \text{div } 2$]*)
(insert $\langle p > 1 \rangle$, auto simp: numeral-3-eq-3 power2-eq-square power-diff algebra-simps elim!: evenE)
also have $?S = (\sum i \mid \text{real}(p \wedge (2 * i + 2)) \leq x. (1 / \text{real } p \wedge 2) \wedge i)$
by (*subst power-mult*) (*simp-all add: algebra-simps power-divide*)
also have $\dots \leq (\sum i. (1 / \text{real } p \wedge 2) \wedge i)$
using *exponents-le-finite[of p 2 2 x] $\langle p > 1 \rangle$*
by (*intro sum-le-suminf*) (*auto simp: summable-geometric-iff*)
also have $\dots = \text{real } p \wedge 2 / (\text{real } p \wedge 2 - 1)$
using $\langle p > 1 \rangle$ **by** (*subst suminf-geometric*) (*auto simp: field-simps*)
also have $\dots / \text{real } p \wedge 2 = 1 / (\text{real } p \wedge 2 - 1)$
using $\langle p > 1 \rangle$ **by** (*simp add: divide-simps*)
finally have $(\sum i \mid 0 < i \wedge \text{even } i \wedge \text{real}(p \wedge i) \leq x. (1 / \text{real } p) \wedge i) \leq$
 $1 / (\text{real } p \wedge 2 - 1)$ **(is ?lhs \leq ?rhs)**
using $\langle p > 1 \rangle$ **by** (*simp add: divide-right-mono*)
thus $\ln(\text{real } p) * ?lhs \leq \ln(\text{real } p) / (\text{real } p \wedge 2 - 1)$
using $\langle p > 1 \rangle$ **by** (*simp add: divide-simps*)
qed
also have $\dots = (\sum_a p \mid \text{prime } p \wedge \text{real } p \leq x. \ln(\text{real } p) / (\text{real } p \wedge 2 - 1))$
using *finite* **by** (*intro infsetsum-finite [symmetric]*) *auto*
also have $\dots \leq (\sum_a p \mid \text{prime } p. \ln(\text{real } p) / (\text{real } p \wedge 2 - 1))$
using *eval-fds-logderiv-zeta-real[of 2] finite*
by (*intro infsetsum-mono-neutral-left divide-nonneg-pos*) (*auto simp: dest: prime-gt-1-nat*)
also have $\dots = -\text{Re}(\text{deriv } \text{zeta}(\text{of-real } 2) / \text{zeta}(\text{of-real } 2))$
by (*subst eval-fds-logderiv-zeta-real*) *auto*
also have $\dots = (-\text{Re}(\text{deriv } \text{zeta } 2)) * (6 / \text{pi}^2)$
by (*simp add: zeta-even-numeral*)
also have $\dots < 1 * (6 / \text{pi}^2)$
using *deriv-zeta-2-bound* **by** (*intro mult-strict-right-mono*) *auto*

also have $3 / 2 * \dots = 9 / pi^2$ by simp
 finally show ?thesis by simp
 qed

We now consider the equation

$$\ln(n!) = \sum_{k \leq n} \Lambda(k) \left\lfloor \frac{n}{k} \right\rfloor$$

and estimate both sides in different ways. The left-hand-side can be estimated using Stirling's formula, and we can simplify the right-hand side to

$$\sum_{k \leq n} \Lambda(k) \left\lfloor \frac{n}{k} \right\rfloor = \sum_{p^i \leq n, i \geq 1} \ln p \left\lfloor \frac{n}{p^i} \right\rfloor$$

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: $\mathfrak{M}(n) - \ln(n) \in (-1 - \frac{9}{\pi^2}; \ln 4]$

theorem *mertens-bound-strong*:

fixes $n :: nat$ **assumes** $n: n > 0$

shows $\mathfrak{M} n - \ln n \in \{-1 - 9 / pi^2 .. \ln 4\}$

proof (cases $n \geq 3$)

case *False*

with n **consider** $n = 1 \mid n = 2$ **by** *force*

thus ?thesis

proof *cases*

assume [*simp*]: $n = 1$

have $-1 + (-9 / pi^2) < 0$

by (*intro add-neg-neg divide-neg-pos*) *auto*

thus ?thesis by simp

next

assume [*simp*]: $n = 2$

have *eq*: $\mathfrak{M} n - \ln n = -\ln 2 / 2$ **by** (*simp add: eval- \mathfrak{M}*)

have $-1 - 9 / pi^2 + \ln 2 / 2 \leq -1 - 9 / 4 + 25 / 36 / 2$

using *pi-less-4 ln2-le-25-over-36*

by (*intro diff-mono add-mono divide-left-mono divide-right-mono power-mono*)

auto

also have $\dots < 0$ **by** *simp*

finally have $-\ln 2 / 2 > -1 - 9 / pi^2$ **by** *simp*

moreover {

have $-\ln 2 / 2 \leq (0 :: real)$ **by** (*intro divide-nonpos-pos*) *auto*

also have $\dots \leq \ln 4$ **by** *simp*

finally have $-\ln 2 / 2 \leq \ln 4$ **by** *simp*

}

ultimately show ?thesis **unfolding** *eq* **by** *simp*

qed

```

next
case True
hence n: n ≥ 3 by simp
have finite: finite {(p, i). prime p ∧ i ≥ 1 ∧ p ^ i ≤ n}
proof (rule finite-subset)
  show {(p, i). prime p ∧ i ≥ 1 ∧ p ^ i ≤ n}
    ⊆ {..nat [root 1 (real n)]} × {..nat [log 2 (real n)]}
  using primepows-le-subset[of real n 1] n unfolding of-nat-le-iff by auto
qed auto

define r where r = prime-sum-upto (λp. ln (real p) * frac (real n / real p)) n
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 ln-3: 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 / pi2
  unfolding R'-def using mertens-remainder-aux-bound[of n] by simp
finally have R / n < 9 / pi2 .
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 / pi2} by simp

have ln (fact n :: real) ≤ ln (2 * pi * n) / 2 + n * ln n - n + 1 / (12 * n)
  using ln-fact-bounds(2)[of n] n by simp
also have ... / n - ln 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 *
32)

```

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 $\dots \leq -1 + (25 / 36 + 25 / 18) / (2 * 3) + (25 / 18 / 3) / 2 + 1 / (12 * 3^2)$
using *ln-pi ln2-le-25-over-36 ln-3* **by** (*intro add-mono divide-left-mono divide-right-mono*) *auto*
also have $\dots \leq 0$ **by** *simp*
finally have $\ln n - \ln (\text{fact } n) / n \geq 0$ **using** *n* **by** (*simp add: divide-right-mono*)
have $-\ln (\text{fact } n) \leq -\ln (2 * \text{pi} * n) / 2 - n * \ln n + n$
using *ln-fact-bounds(1)[of n] n* **by** *simp*
also have $\ln n + \dots / n = -\ln (2 * \text{pi}) / (2 * n) - (\ln n / n) / 2 + 1$
using *n* **by** (*simp add: field-simps ln-mult*)
also have $\dots \leq 0 - 0 + 1$
using *pi-gt3 n* **by** (*intro add-mono diff-mono*) *auto*
finally have *upper: ln n - ln (fact n) / n ≤ 1*
using *n* **by** (*simp add: divide-right-mono*)
with $\langle \ln n - \ln (\text{fact } n) / n \geq 0 \rangle$ **have** *fact-bounds: ln n - ln (fact n) / n ∈ {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* *∅-def* *prime-sum-upto-def*
by (*intro sum-mono mult-left-mono*) (*auto simp: dest: prime-gt-0-nat*)
also have $\dots = \emptyset$ **by** (*simp add: ∅-def*)
also have $\dots < \ln 4 * n$ **using** *n* **by** (*intro ∅-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-gt-0-nat*)
ultimately have *r-bounds: r / n ∈ {0..<ln 4}* **by** *simp*

have $\ln (\text{fact } n :: \text{real}) = \text{sum-upto } (\lambda k. \text{mangoldt } k * \text{real } (n \text{ div } k)) (\text{real } n)$
by (*simp add: ln-fact-conv-sum-upto-mangoldt*)
also have $\dots = (\sum (p,i) \mid \text{prime } p \wedge i > 0 \wedge \text{real } (p \wedge i) \leq \text{real } n. \ln (\text{real } p) * \text{real } (n \text{ div } (p \wedge i)))$
by (*intro sum-upto-primepows*) (*auto simp: mangoldt-non-primepow*)
also have $\{(p, i). \text{prime } p \wedge i > 0 \wedge \text{real } (p \wedge i) \leq \text{real } n\} =$
 $\{(p, i). \text{prime } p \wedge i = 1 \wedge p \leq n\} \cup$
 $\{(p, i). \text{prime } p \wedge i > 1 \wedge (p \wedge i) \leq n\}$ **unfolding** *of-nat-le-iff*
by (*auto simp: not-less le-Suc-eq*)
also have $(\sum (p,i) \in \dots \ln (\text{real } p) * \text{real } (n \text{ div } (p \wedge i))) =$
 $(\sum (p,i) \mid \text{prime } p \wedge i = 1 \wedge p \leq n. \ln (\text{real } p) * \text{real } (n \text{ div } (p \wedge i)))$

+ *R*
(is - = ?S + -)
by (*subst sum.union-disjoint*) (*auto intro!: finite-subset[OF - finite] simp: R-def*)
also have *?S = 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 - λp. (p, 1) fst]*) *auto*
also have $\dots = \text{prime-sum-upto } (\lambda p. \ln (\text{real } p) * \text{real } n / \text{real } p) n - r$
unfolding *r-def* *prime-sum-upto-def* *sum-subtractf[symmetric]* **using** *n*

by (intro sum.cong) (auto simp: frac-def real-of-nat-div algebra-simps)
 also have prime-sum-upto ($\lambda p. \ln (\text{real } p) * \text{real } n / \text{real } p$) $n = n * \mathfrak{M} n$
 by (simp add: primes-M-def sum-distrib-left sum-distrib-right prime-sum-upto-def
 field-simps)
 finally have $\mathfrak{M} n - \ln n = \ln (\text{fact } n) / n - \ln n + r / n - R / n$
 using n by (simp add: field-simps)
 hence $\ln n - \mathfrak{M} n = \ln n - \ln (\text{fact } n) / n - r / n + R / n$
 by simp
 with fact-bounds r-bounds R-bounds show $\mathfrak{M} n - \ln n \in \{-1 - 9 / \pi^2 <.. \ln 4\}$
 by simp
 qed

As a simple corollary, we obtain a similar bound on the reals.

lemma *mertens-bound-real-strong*:

fixes $x :: \text{real}$ assumes $x: x \geq 1$
 shows $\mathfrak{M} x - \ln x \in \{-1 - 9 / \pi^2 - \ln (1 + \text{frac } x / \text{real } (\text{nat } \lfloor x \rfloor)) <.. \ln 4\}$

proof –

have $\mathfrak{M} x - \ln x \leq \mathfrak{M} (\text{real } (\text{nat } \lfloor x \rfloor)) - \ln (\text{real } (\text{nat } \lfloor x \rfloor))$
 using *assms* by simp
 also have $\dots \leq \ln 4$
 using *mertens-bound-strong*[of $\text{nat } \lfloor x \rfloor$] *assms* by simp
 finally have $\mathfrak{M} x - \ln x \leq \ln 4$.

from *assms* have *pos*: $\text{real-of-int } \lfloor x \rfloor \neq 0$ by *linarith*

have $\text{frac } x / \text{real } (\text{nat } \lfloor x \rfloor) \geq 0$

using *assms* by (intro *divide-nonneg-pos*) *auto*

moreover have $\text{frac } x / \text{real } (\text{nat } \lfloor x \rfloor) \leq 1 / 1$

using *assms* *frac-lt-1*[of x] by (intro *frac-le*) *auto*

ultimately have $*$: $\text{frac } x / \text{real } (\text{nat } \lfloor x \rfloor) \in \{0..1\}$ by *auto*

have $\ln x - \ln (\text{real } (\text{nat } \lfloor x \rfloor)) = \ln (x / \text{real } (\text{nat } \lfloor x \rfloor))$

using *assms* by (*subst ln-div*) *auto*

also have $x / \text{real } (\text{nat } \lfloor x \rfloor) = 1 + \text{frac } x / \text{real } (\text{nat } \lfloor x \rfloor)$

using *assms pos* by (*simp add: frac-def field-simps*)

finally have $\mathfrak{M} x - \ln x > -1 - 9 / \pi^2 - \ln (1 + \text{frac } x / \text{real } (\text{nat } \lfloor x \rfloor))$

using *mertens-bound-strong*[of $\text{nat } \lfloor x \rfloor$] x by *simp*

with $\langle \mathfrak{M} x - \ln x \leq \ln 4 \rangle$ show *?thesis* by *simp*

qed

We weaken this estimate a bit to obtain nicer bounds:

lemma *mertens-bound-real'*:

fixes $x :: \text{real}$ assumes $x: x \geq 1$

shows $\mathfrak{M} x - \ln x \in \{-2 <.. 25 / 18\}$

proof –

have $\mathfrak{M} x - \ln x \leq \ln 4$

using *mertens-bound-real-strong*[of x] x by *simp*

also have $\dots \leq 25 / 18$

using *ln-realpow*[of 2 2] *ln2-le-25-over-36* by *simp*

finally have $\mathfrak{M} x - \ln x \leq 25 / 18$.

```

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}
  using ln-approx-bounds[of 11 1] by (simp add: eval-nat-numeral)

```

— Choosing the lower bound -2 is somewhat arbitrary here; it is a trade-off between getting a reasonably tight bound and having to make lots of case distinctions. To get -2 as a lower bound, we have to show the cases up to $x = 11$ by case distinction,

```

have  $\mathfrak{M} x - \ln x > -2$ 
proof (cases  $x \geq 11$ )
  case False
    hence  $x \in \{1..<2\} \vee x \in \{2..<3\} \vee x \in \{3..<5\} \vee x \in \{5..<7\} \vee x \in \{7..<11\}$ 
      using  $x$  by force
    thus ?thesis
    proof (elim disjE)
      assume  $x: x \in \{1..<2\}$ 
      hence  $\ln x - \mathfrak{M} x \leq \ln 2 - 0$ 
        by (intro diff-mono) auto
      also have  $\dots < 2$  using ln2-le-25-over-36 by simp
      finally show ?thesis by simp
    next
      assume  $x: x \in \{2..<3\}$ 
      hence [simp]:  $\lfloor x \rfloor = 2$  by (intro floor-unique) auto
      from  $x$  have  $\ln x - \mathfrak{M} x \leq \ln 3 - \ln 2 / 2$ 
        by (intro diff-mono) (auto simp: eval- $\mathfrak{M}$ )
      also have  $\dots = \ln (9 / 2) / 2$  using ln-realpow[of 3 2] by (simp add: ln-div)
      also have  $\dots < 2$  using ln-approx-bounds[of 9 / 2 1] by (simp add:
eval-nat-numeral)
      finally show ?thesis by simp
    next
      assume  $x: x \in \{3..<5\}$ 
      hence  $\mathfrak{M} 3 = \mathfrak{M} x$ 
        unfolding primes-M-def
        by (intro prime-sum-upto-eqI'[where  $a' = 3$  and  $b' = 4$ ])
          (auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)
      also have  $\mathfrak{M} 3 = \ln 2 / 2 + \ln 3 / 3$ 
        by (simp add: eval- $\mathfrak{M}$  eval-nat-numeral mark-out-code)
      finally have [simp]:  $\mathfrak{M} x = \ln 2 / 2 + \ln 3 / 3$  ..
      from  $x$  have  $\ln x - \mathfrak{M} x \leq \ln 5 - (\ln 2 / 2 + \ln 3 / 3)$ 

```

```

    by (intro diff-mono) auto
    also have ... < 2 using ln2 ln3 ln5 by simp
    finally show ?thesis by simp
next
assume x: x ∈ {5..<7}
hence ℳ 5 = ℳ x
  unfolding primes-M-def
  by (intro prime-sum-upto-eqI'[where a' = 5 and b' = 6])
    (auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)
also have ℳ 5 = ln 2 / 2 + ln 3 / 3 + ln 5 / 5
  by (simp add: eval-ℳ eval-nat-numeral mark-out-code)
finally have [simp]: ℳ x = ln 2 / 2 + ln 3 / 3 + ln 5 / 5 ..
from x have ln x - ℳ x ≤ ln 7 - (ln 2 / 2 + ln 3 / 3 + ln 5 / 5)
  by (intro diff-mono) auto
also have ... < 2 using ln2 ln3 ln5 ln7 by simp
finally show ?thesis by simp
next
assume x: x ∈ {7..<11}
hence ℳ 7 = ℳ x
  unfolding primes-M-def
  by (intro prime-sum-upto-eqI'[where a' = 7 and b' = 10])
    (auto simp: nat-le-iff le-numeral-iff nat-eq-iff floor-eq-iff)
also have ℳ 7 = ln 2 / 2 + ln 3 / 3 + ln 5 / 5 + ln 7 / 7
  by (simp add: eval-ℳ eval-nat-numeral mark-out-code)
finally have [simp]: ℳ x = ln 2 / 2 + ln 3 / 3 + ln 5 / 5 + ln 7 / 7 ..
from x have ln x - ℳ x ≤ ln 11 - (ln 2 / 2 + ln 3 / 3 + ln 5 / 5 + ln 7
/ 7)
  by (intro diff-mono) auto
also have ... < 2 using ln2 ln3 ln5 ln7 ln11 by simp
finally show ?thesis by simp
qed
next
case True
have ln x - ℳ x ≤ 1 + 9/pi^2 + ln (1 + frac x / real (nat ⌊x⌋))
  using mertens-bound-real-strong[of x] x by simp
also have 1 + frac x / real (nat ⌊x⌋) ≤ 1 + 1 / 11
  using True frac-lt-1[of x] by (intro add-mono frac-le) auto
hence ln (1 + frac x / real (nat ⌊x⌋)) ≤ ln (1 + 1 / 11)
  using x by (subst ln-le-cancel-iff) (auto intro!: add-pos-nonneg)
also have ... = ln (12 / 11) by simp
also have ... ≤ 1585 / 18216
  using ln-approx-bounds[of 12 / 11 1] by (simp add: eval-nat-numeral)
also have 9 / pi ^ 2 ≤ 9 / 3.141592653588 ^ 2
  using pi-approx by (intro divide-left-mono power-mono mult-pos-pos) auto
also have 1 + ... + 1585 / 18216 < 2
  by (simp add: power2-eq-square)
finally show ?thesis by simp
qed
with ⟨ℳ x - ln x ≤ 25 / 18⟩ show ?thesis by simp

```

qed

corollary *mertens-first-theorem*:

fixes $x :: \text{real}$ **assumes** $x: x \geq 1$

shows $|\mathfrak{M} x - \ln x| < 2$

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 = $1 - \ln(\ln 2) + \text{integral } \{2..\} (\lambda t. (\mathfrak{M} t - \ln t) / (t * \ln t ^ \wedge 2))$

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 $a: (a::\text{real}) > 1$

shows *set-integrable lborel* $\{a<..\}$ $(\lambda t. 1 / (t * \ln t ^ \wedge 2))$ (**is** *?th1*)

and *set-lebesgue-integral lborel* $\{a<..\}$ $(\lambda t. 1 / (t * \ln t ^ \wedge 2)) = 1 / \ln a$ (**is** *?th2*)

and $((\lambda t. 1 / (t * (\ln t)^2)) \text{ has-integral } 1 / \ln a) \{a<..\}$ (**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*: $((\lambda x. -1 / \ln x) \text{ has-real-derivative } 1 / (x * (\ln x)^2)) (at\ x)$ **if** $x > a$ **for** x

using *that a* **by** (*auto intro!*: *derivative-eq-intros simp: power2-eq-square field-simps*)

have *lim1*: $((\lambda x. -1 / \ln x) \circ \text{real-of-ereal}) \longrightarrow -(1 / \ln a)$ (*at-right (ereal a)*)

unfolding *ereal-tendsto-simps* **using** a **by** (*real-asymp simp: field-simps*)

have *lim2*: $((\lambda x. -1 / \ln x) \circ \text{real-of-ereal}) \longrightarrow 0$ (*at-left* ∞)

unfolding *ereal-tendsto-simps* **using** a **by** (*real-asymp simp: field-simps*)

have *set-integrable lborel (einterval a* $\infty)$ $(\lambda t. 1 / (t * \ln t ^ \wedge 2))$

by (*rule interval-integral-FTC-nonneg*[*OF - deriv cont - lim1 lim2*]) (*use a in auto*)

thus *?th1* **by** *simp*

have *interval-lebesgue-integral lborel (ereal a)* $\infty (\lambda t. 1 / (t * \ln t ^ \wedge 2)) = 0 - (-1 / \ln a)$

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
  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  $\equiv$  ( $\lambda t. (\mathfrak{M} t - \ln t) / (t * \ln t ^ 2)$ )
shows integrable-meissel-mertens: set-integrable lborel {a<..} r
  and meissel-mertens-integral-le: norm (integral {a<..} r)  $\leq$  2 / ln a
proof –
  have *: (( $\lambda t. 2 * (1 / (t * \ln t ^ 2))$ ) has-integral 2 * (1 / ln a)) {a<..}
  using assms by (intro has-integral-mult-right integral-one-over-x-ln-x-squared)
auto
  show set-integrable lborel {a<..} r unfolding set-integrable-def
proof (rule Bochner-Integration.integrable-bound[OF - - AE-I2])
  have integrable lborel ( $\lambda t::real. \text{indicator } \{a<..\} t * (2 * (1 / (t * \ln t ^ 2)))$ )
  using integrable-mult-right[of 2,
    OF integral-one-over-x-ln-x-squared(1)[of a, unfolded set-integrable-def]]
  assms
  by (simp add: algebra-simps)
  thus integrable lborel ( $\lambda t::real. \text{indicator } \{a<..\} t *_{\mathbb{R}} (2 / (t * \ln t ^ 2))$ )
  by simp
  fix x :: real
  show norm (indicat-real {a<..} x *_R r x)  $\leq$ 
    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)  $\leq$  integral {a<..} ( $\lambda x. 2 * (1 / (x * \ln x ^ 2))$ )
proof (rule integral-norm-bound-integral)
  show ( $\lambda x. 2 * (1 / (x * (\ln x)^2))$ ) integrable-on {a<..}

```

using * **by** (*simp add: has-integral-iff*)
fix x **assume** $x \in \{a<..\}$
hence $\text{norm } (r x) \leq 2 / (x * (\ln x)^2)$
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 $\text{norm } (r x) \leq 2 * (1 / (x * \ln x ^ 2))$ **by** *simp*
qed
also have $\dots = 2 / \ln a$
using * **by** (*simp add: has-integral-iff*)
finally show $\text{norm } (\text{integral } \{a<..\} r) \leq 2 / \ln a$.
qed

lemma *integrable-on-meissel-mertens*:

assumes $A \subseteq \{1..\}$ $\text{Inf } A > 1$ $A \in \text{sets borel}$
shows $(\lambda t. (\mathfrak{M} t - \ln t) / (t * \ln t ^ 2))$ *integrable-on* A
proof –
from *assms* **obtain** x **where** $x: 1 < x < \text{Inf } A$
using *dense* **by** *blast*
from *assms* **have** *bdd-below* A **by** (*intro bdd-belowI[of - 1]*) *auto*
hence $A \subseteq \{\text{Inf } A..\}$ **by** (*auto simp: cInf-lower*)
also have $\dots \subseteq \{x<..\}$ **using** x **by** *auto*
finally have *: $A \subseteq \{x<..\}$.
have *set-integrable lborel* A $(\lambda t. (\mathfrak{M} t - \ln t) / (t * \ln t ^ 2))$
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)*)
qed

lemma *meissel-mertens-bounds*: $|\text{meissel-mertens} - 1 + \ln (\ln 2)| \leq 2 / \ln 2$

proof –
have *: $\{2..\} - \{2<..\} = \{2::\text{real}\}$ **by** *auto*
also have *negligible* \dots **by** *simp*
finally have $\text{integral } \{2..\} (\lambda t. (\mathfrak{M} t - \ln t) / (t * (\ln t)^2)) =$
 $\text{integral } \{2<..\} (\lambda t. (\mathfrak{M} t - \ln t) / (t * (\ln t)^2))$
by (*intro sym[OF integral-subset-negligible]*) *auto*
also have $\text{norm } \dots \leq 2 / \ln 2$
by (*rule meissel-mertens-integral-le*) *auto*
finally show $|\text{meissel-mertens} - 1 + \ln (\ln 2)| \leq 2 / \ln 2$
by (*simp add: meissel-mertens-def*)
qed

Finally, obtaining Mertens' second theorem from the first one is nothing but a routine summation by parts, followed by a use of the above bound:

theorem *mertens-second-theorem*:

defines $f \equiv \text{prime-sum-upto } (\lambda p. 1 / p)$
shows $\bigwedge x. x \geq 2 \implies |f x - \ln (\ln x) - \text{meissel-mertens}| \leq 4 / \ln x$
and $(\lambda x. f x - \ln (\ln x) - \text{meissel-mertens}) \in O(\lambda x. 1 / \ln x)$
proof –
define r **where** $r = (\lambda t. (\mathfrak{M} t - \ln t) / (t * \ln t ^ 2))$

```

{
  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<..x}. 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 {}])
    (auto intro!: continuous-intros derivative-eq-intros simp: power2-eq-square
      simp flip: has-field-derivative-iff-has-vector-derivative)
  also have ℳ x * (1 / ln x) - ℳ 2 * (1 / ln 2) -
    (∑ n∈real - ' {2<..x}. ind prime n * (ln n / n) * (1 / ln n)) =
    ℳ x / ln x - (∑ n∈insert 2 (real - ' {2<..x}). ind prime n * (ln n /
n) * (1 / ln n))
    (is - = - - ?S)
  by (subst sum.insert)
    (auto simp: primes-M-def finite-vimage-real-of-nat-greaterThanAtMost
eval-prime-sum-upto)
  also have ?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 ((λt. -ℳ t / (t * ln t ^ 2)) has-integral (ℳ x / ln x - f x))
{2..x}
  by simp
  from has-integral-neg[OF this]
  have ((λt. ℳ t / (t * ln t ^ 2)) has-integral (f x - ℳ x / ln x)) {2..x} by
simp
  hence ((λt. ℳ t / (t * ln t ^ 2) - 1 / (t * ln t)) has-integral
(f x - ℳ 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-field-derivative-iff-has-vector-derivative
      intro!: derivative-eq-intros)
  also have ?this ↔ (r has-integral (f x - ℳ 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 Rℳ where Rℳ = (λx. ℳ x - ln x)
have ℳ: ℳ x = ln x + Rℳ x for x by (simp add: Rℳ-def)
define I where I = (λx. integral {x..} r)
define C where C = (1 - ln (ln 2) + I 2)
have C-altdef: C = meissel-mertens
  by (simp add: I-def r-def C-def meissel-mertens-def)

show bound: |f x - ln (ln x) - meissel-mertens| ≤ 4 / ln x if x: x ≥ 2 for x
proof (cases x = 2)
  case True

```

hence $|f x - \ln (\ln x) - \text{meissel-mertens}| = |1 / 2 - \ln (\ln 2) - \text{meissel-mertens}|$
by (*simp add: f-def eval-prime-sum-upto*)
also have $\dots \leq 2 / \ln 2 + 1 / 2$
using *meissel-mertens-bounds* **by** *linarith*
also have $\dots \leq 2 / \ln 2 + 2 / \ln 2$ **using** *ln2-le-25-over-36*
by (*intro add-mono divide-left-mono*) *auto*
finally show *?thesis* **using** *True* **by** *simp*
next
case *False*
hence $x: x > 2$ **using** x **by** *simp*
have $\text{integral } \{2..x\} r + I x = \text{integral } (\{2..x\} \cup \{x..}) r$ **unfolding** *I-def r-def*
using x
by (*intro integral-Un [symmetric] integrable-on-meissel-mertens*) (*auto simp: max-def r-def*)
also have $\{2..x\} \cup \{x..} = \{2..}$ **using** x **by** *auto*
finally have $*$: $\text{integral } \{2..x\} r = I 2 - I x$ **unfolding** *I-def* **by** *simp*
have $\text{eq: } f x - \ln (\ln x) - C = R_{\mathfrak{M}} x / \ln x - I x$
using $\text{integral}[OF x] x$ **by** (*auto simp: C-def field-simps* \mathfrak{M} *has-integral-iff **)
also have $|\dots| \leq |R_{\mathfrak{M}} x / \ln x| + \text{norm } (I x)$
unfolding *real-norm-def* **by** (*rule abs-triangle-ineq4*)
also have $|R_{\mathfrak{M}} x / \ln x| \leq 2 / |\ln x|$
unfolding *R_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 $\text{norm } \dots \leq 2 / \ln x$
using *meissel-mertens-integral-le[of x]* x **by** (*simp add: r-def*)
finally show $|f x - \ln (\ln x) - \text{meissel-mertens}| \leq 4 / \ln x$
using x **by** (*simp add: C-altdef*)
qed

have $(\lambda x. f x - \ln (\ln x) - C) \in O(\lambda x. 1 / \ln x)$
proof (*intro landau-o.bigI[of 4] eventually-mono[OF eventually-ge-at-top[of 2]]*)
fix $x :: \text{real}$ **assume** $x: x \geq 2$
with $\text{bound}[OF x]$ **show** $\text{norm } (f x - \ln (\ln x) - C) \leq 4 * \text{norm } (1 / \ln x)$
by (*simp add: C-altdef*)
qed (*auto intro!: add-pos-nonneg*)
thus $(\lambda x. f x - \ln (\ln x) - \text{meissel-mertens}) \in O(\lambda x. 1 / \ln x)$
by (*simp add: C-altdef*)
qed

corollary *prime-harmonic-asymp-equiv: prime-sum-upto* $(\lambda p. 1 / p) \sim[\text{at-top}] (\lambda x. \ln (\ln x))$
proof –
define f **where** $f = \text{prime-sum-upto } (\lambda p. 1 / p)$
have $(\lambda x. f x - \ln (\ln x) - \text{meissel-mertens} + \text{meissel-mertens}) \in o(\lambda x. \ln (\ln x))$

unfolding *f-def*
by (*rule sum-in-smallo*[*OF landau-o.big-small-trans*[*OF mertens-second-theorem(2)*]])
real-asymp+
hence $(\lambda x. f x - \ln (\ln x)) \in o(\lambda x. \ln (\ln x))$
by *simp*
thus *?thesis unfolding f-def*
by (*rule smallo-imp-asymp-equiv*)
qed

As a corollary, we get the divergence of the prime harmonic series.

corollary *prime-harmonic-diverges: filterlim (prime-sum-upto ($\lambda p. 1 / p$)) at-top at-top*
using *asymp-equiv-symI*[*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

- [1] T. M. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer-Verlag, 1976.
- [2] J. Avigad, K. Donnelly, D. Gray, and P. Raff. A formally verified proof of the prime number theorem. *ACM Trans. Comput. Logic*, 9(1), Dec. 2007.
- [3] M. Carneiro. Formalization of the prime number theorem and dirichlet’s theorem. In *Proceedings of the 9th Conference on Intelligent Computer Mathematics (CICM 2016)*, pages 10–13, 2016.
- [4] O. Forster. Analytic Number Theory (lecture notes). http://www.mathematik.uni-muenchen.de/~forster/v/ann/anntth_all.pdf.
- [5] J. Harrison. Formalizing an analytic proof of the Prime Number Theorem (dedicated to Mike Gordon on the occasion of his 60th birthday). *Journal of Automated Reasoning*, 43:243–261, 2009.
- [6] D. Newman. *Analytic Number Theory*. Number 177 in Graduate Texts in Mathematics. Springer, 1998.