

# The Hurwitz and Riemann $\zeta$ functions

Manuel Eberl

May 26, 2024

## Abstract

This entry builds upon the results about formal and analytic Dirichlet series to define the Hurwitz  $\zeta$  function  $\zeta(a, s)$  and, based on that, the Riemann  $\zeta$  function  $\zeta(s)$ . This is done by first defining them for  $\Re(z) > 1$  and then successively extending the domain to the left using the Euler–MacLaurin formula.

Apart from the most basic facts such as analyticity, the following results are provided:

- the Stieltjes constants and the Laurent expansion of  $\zeta(s)$  at  $s = 1$
- the non-vanishing of  $\zeta(s)$  for  $\Re(s) \geq 1$
- the relationship between  $\zeta(a, s)$  and  $\Gamma$
- the special values at negative integers and positive even integers
- Hurwitz’s formula and the reflection formula for  $\zeta(s)$
- the Hadjicostas–Chapman formula [3, 4]

The entry also contains Euler’s analytic proof of the infinitude of primes, based on the fact that  $\zeta(s)$  has a pole at  $s = 1$ .

# Contents

<b>1</b>	<b>Various preliminary material</b>	<b>3</b>
1.1	Facts about limits . . . . .	3
1.2	Various facts about integrals . . . . .	7
1.3	Uniform convergence of integrals . . . . .	7
1.4	Other material . . . . .	19
<b>2</b>	<b>The Hurwitz and Riemann <math>\zeta</math> functions</b>	<b>20</b>
2.1	Preliminary facts . . . . .	20
2.2	Definitions . . . . .	23
2.3	Connection to Dirichlet series . . . . .	37
2.4	The non-vanishing of $\zeta$ for $\Re(s) \geq 1$ . . . . .	42
2.5	Special values of the $\zeta$ functions . . . . .	47
2.6	Integral relation between $\Gamma$ and $\zeta$ function . . . . .	51
2.7	An analytic proof of the infinitude of primes . . . . .	57
2.8	The periodic zeta function . . . . .	59
2.9	Hurwitz's formula . . . . .	62
2.10	More functional equations . . . . .	96
<b>3</b>	<b>The Laurent series expansion of <math>\zeta</math> at 1</b>	<b>100</b>
3.1	Definition of the Stieltjes constants . . . . .	100
3.2	Proof of the Laurent expansion . . . . .	102
<b>4</b>	<b>The Hadjicostas–Chapman formula</b>	<b>112</b>
4.1	The real case . . . . .	112
4.2	Analyticity of the integral . . . . .	117
4.3	Analytic continuation and main result . . . . .	124

# 1 Various preliminary material

**theory** *Zeta-Library*

**imports**

*HOL-Complex-Analysis.Complex-Analysis*

*HOL-Real-Asymp.Real-Asymp*

*Dirichlet-Series.Dirichlet-Series-Analysis*

**begin**

## 1.1 Facts about limits

**lemma** *at-within-altdef*:

*at x within A = (INF S ∈ {S. open S ∧ x ∈ S}. principal (S ∩ (A - {x})))*

**unfolding** *at-within-def nhds-def inf-principal [symmetric]*

**by** (*subst INF-inf-distrib [symmetric]*) (*auto simp: INF-constant*)

**lemma** *tendsto-at-left-real-sequentially*:

**fixes** *f :: real ⇒ 'b::first-countable-topology*

**assumes** *\*: ∧X. filterlim X (at-left c) sequentially ⇒ (λn. f (X n)) → y*

**shows** (*f → y*) (*at-left c*)

**proof** –

**obtain** *A where A: decseq A open (A n) y ∈ A n nhds y = (INF n. principal (A n)) for n*

**by** (*rule nhds-countable[of y]*) (*rule that*)

**have**  $\forall m. \exists d < c. \forall x \in \{d < .. < c\}. f x \in A m$

**proof** (*rule ccontr*)

**assume**  $\neg (\forall m. \exists d < c. \forall x \in \{d < .. < c\}. f x \in A m)$

**then obtain** *m where \*\*: ∧d. d < c ⇒ ∃x ∈ {d < .. < c}. f x ∉ A m*

**by** *auto*

**have**  $\exists X. \forall n. (f (X n) \notin A m \wedge X n < c) \wedge X (Suc n) > c - \max 0 ((c - X n) / 2)$

**proof** (*intro dependent-nat-choice, goal-cases*)

**case** *1*

**from** *\*\*[of c - 1]* **show** *?case by auto*

**next**

**case** (*2 x n*)

**with** *\*\*[of c - max 0 (c - x) / 2]* **show** *?case by force*

**qed**

**then obtain** *X where X: ∧n. f (X n) ∉ A m ∧n. X n < c ∧n. X (Suc n) > c - max 0 ((c - X n) / 2)*

**by** *auto (metis diff-gt-0-iff-gt half-gt-zero-iff max.absorb3 max.commute)*

**have** *X-ge: X n ≥ c - (c - X 0) / 2 ^ n for n*

**proof** (*induction n*)

**case** (*Suc n*)

**have**  $c - (c - X 0) / 2 ^ Suc n = c - (c - (c - (c - X 0) / 2 ^ n)) / 2$

**by** *simp*

**also have**  $c - (c - (c - (c - X 0) / 2 ^ n)) / 2 \leq c - (c - X n) / 2$

**by** (*intro diff-left-mono divide-right-mono Suc diff-right-mono*) *auto*

**also have**  $\dots = c - \max 0 ((c - X n) / 2)$

**using**  $X[of\ n]$  **by** (*simp add: max-def*)  
**also have**  $\dots < X\ (Suc\ n)$   
**using**  $X[of\ n]$  **by** *simp*  
**finally show** *?case by linarith*  
**qed** *auto*

**have**  $X \longrightarrow c$   
**proof** (*rule tendsto-sandwich*)  
**show** *eventually*  $(\lambda n. X\ n \leq c)$  *sequentially*  
**using**  $X$  **by** (*intro always-eventually*) (*auto intro!: less-imp-le*)  
**show** *eventually*  $(\lambda n. X\ n \geq c - (c - X\ 0) / 2 \wedge n)$  *sequentially*  
**using**  $X$ -ge **by** (*intro always-eventually*) *auto*  
**qed** *real-asymp+*  
**hence** *filterlim*  $X$  (*at-left*  $c$ ) *sequentially*  
**by** (*rule tendsto-imp-filterlim-at-left*)  
*(use  $X$  in  $\langle auto\ intro!: always-eventually\ less-imp-le \rangle$ )*  
**from** *topological-tendstoD*[*OF* \*[*OF* *this*]  $A(2, 3)$ , *of*  $m$ ]  $X(1)$  **show** *False*  
**by** *auto*  
**qed**

**then obtain**  $d$  **where**  $d\ m < c\ x \in \{d\ m < .. < c\} \implies f\ x \in A\ m$  **for**  $m\ x$   
**by** *metis*  
**have** *\*\*\*: at-left*  $c = (INF\ S \in \{S. open\ S \wedge c \in S\}. principal\ (S \cap \{.. < c\}))$   
**by** (*simp add: at-within-altdef*)  
**from**  $d$  **show** *?thesis*  
**unfolding** *\*\*\**  $A(1,2)$  **by** (*intro filterlim-base*[*of* -  $\lambda m. \{d\ m < ..\}$ ])  
*auto*  
**qed**

**lemma**  
**shows** *at-right-PInf* [*simp*]: *at-right*  $(\infty :: ereal) = bot$   
**and** *at-left-MInf* [*simp*]: *at-left*  $(-\infty :: ereal) = bot$   
**proof** -  
**have**  $\{(\infty :: ereal) < ..\} = \{\} \{.. < -(\infty :: ereal)\} = \{\}$   
**by** *auto*  
**thus** *at-right*  $(\infty :: ereal) = bot$  *at-left*  $(-\infty :: ereal) = bot$   
**by** (*simp-all add: at-within-def*)  
**qed**

**lemma** *tendsto-at-left-erealI-sequentially*:  
**fixes**  $f :: ereal \Rightarrow 'b :: first-countable-topology$   
**assumes**  $*$ :  $\bigwedge X. filterlim\ X\ (at-left\ c)\ sequentially \implies (\lambda n. f\ (X\ n)) \longrightarrow y$   
**shows**  $(f \longrightarrow y)$  (*at-left*  $c$ )  
**proof** (*cases*  $c$ )  
**case** [*simp*]: *PInf*  
**have**  $((\lambda x. f\ (ereal\ x)) \longrightarrow y)$  *at-top* **using** *assms*  
**by** (*intro tendsto-at-topI-sequentially* *assms*)  
*(simp-all flip: ereal-tendsto-simps add: o-def filterlim-at)*  
**thus** *?thesis*

```

    by (simp add: at-left-PInf filterlim-filtermap)
next
case [simp]: MInf
thus ?thesis by auto
next
case [simp]: (real c')
have ((λx. f (ereal x)) ⟶ y) (at-left c')
proof (intro tendsto-at-left-realI-sequentially assms)
  fix X assume *: filterlim X (at-left c') sequentially
  show filterlim (λn. ereal (X n)) (at-left c) sequentially
  by (rule filterlim-compose[OF - *])
  (simp add: sequentially-imp-eventually-within tendsto-imp-filterlim-at-left)
qed
thus ?thesis
  by (simp add: at-left-ereal filterlim-filtermap)
qed

lemma tendsto-at-right-realI-sequentially:
  fixes f :: real ⇒ 'b::first-countable-topology
  assumes *: ⋀X. filterlim X (at-right c) sequentially ⟹ (λn. f (X n)) ⟶ y
  shows (f ⟶ y) (at-right c)
proof -
  obtain A where A: decseq A open (A n) y ∈ A n nhds y = (INF n. principal
(A n)) for n
  by (rule nhds-countable[of y]) (rule that)

  have ∀ m. ∃ d > c. ∀ x ∈ {c <..

```

```

    using X[of n] by (simp add: field-simps max-def)
  also have ... ≤ c + (c + (X 0 - c) / 2 ^ n - c) / 2
    by (intro add-left-mono divide-right-mono Suc diff-right-mono) auto
  also have ... = c + (X 0 - c) / 2 ^ Suc n
    by simp
  finally show ?case by linarith
qed auto

```

```

have X ⟶ c
proof (rule tendsto-sandwich)
  show eventually (λn. X n ≥ c) sequentially
    using X by (intro always-eventually) (auto intro!: less-imp-le)
  show eventually (λn. X n ≤ c + (X 0 - c) / 2 ^ n) sequentially
    using X-le by (intro always-eventually) auto
qed real-asymp+
hence filterlim X (at-right c) sequentially
  by (rule tendsto-imp-filterlim-at-right)
  (use X in ⟨auto intro!: always-eventually less-imp-le⟩)
from topological-tendstoD[OF *[OF this] A(2, 3), of m] X(1) show False
  by auto
qed

```

```

then obtain d where d: d m > c x ∈ {c<..

```

```

lemma tendsto-at-right-erealI-sequentially:
  fixes f :: ereal ⇒ 'b::first-countable-topology
  assumes *: ⋀X. filterlim X (at-right c) sequentially ⟹ (λn. f (X n)) ⟶ y
  shows (f ⟶ y) (at-right c)
proof (cases c)
  case [simp]: MInf
  have ((λx. f (-ereal x)) ⟶ y) at-top using assms
    by (intro tendsto-at-topI-sequentially assms)
    (simp-all flip: uminus-ereal.simps ereal-tendsto-simps add: o-def filterlim-at)
  thus ?thesis
    by (simp add: at-right-MInf filterlim-filtermap at-top-mirror)
next
  case [simp]: PInf
  thus ?thesis by auto
next
  case [simp]: (real c')
  have ((λx. f (ereal x)) ⟶ y) (at-right c')
  proof (intro tendsto-at-right-realI-sequentially assms)

```

```

fix X assume *: filterlim X (at-right c1) sequentially
show filterlim (λn. ereal (X n)) (at-right c) sequentially
  by (rule filterlim-compose[OF - *])
      (simp add: sequentially-imp-eventually-within tendsto-imp-filterlim-at-right)
qed
thus ?thesis
  by (simp add: at-right-ereal filterlim-filtermap)
qed

```

```

proposition analytic-continuation':
assumes hol: f holomorphic-on S g holomorphic-on S
  and open S and connected S
  and U ⊆ S and ξ ∈ S
  and ξ islimpt U
  and fU0 [simp]: ∧z. z ∈ U ⇒ f z = g z
  and w ∈ S
shows f w = g w
using analytic-continuation[OF holomorphic-on-diff[OF hol] assms(3-7) - assms(9)]
  assms(8)
by simp

```

## 1.2 Various facts about integrals

```

lemma continuous-on-imp-set-integrable-cbox:
fixes h :: 'a :: euclidean-space ⇒ 'b :: euclidean-space
assumes continuous-on (cbox a b) h
shows set-integrable lborel (cbox a b) h
by (simp add: assms borel-integrable-compact set-integrable-def)

```

## 1.3 Uniform convergence of integrals

```

lemma has-absolute-integral-change-of-variables-1':
fixes f :: real ⇒ real and g :: real ⇒ real
assumes S: S ∈ sets lebesgue
  and der-g: ∧x. x ∈ S ⇒ (g has-field-derivative g' x) (at x within S)
  and inj: inj-on g S
shows (λx. |g' x| *R f(g x)) absolutely-integrable-on S ∧
  integral S (λx. |g' x| *R f(g x)) = b
  ⇔ f absolutely-integrable-on (g ' S) ∧ integral (g ' S) f = b
proof -
have (λx. |g' x| *R vec (f(g x)) :: real ^ 1) absolutely-integrable-on S ∧
  integral S (λx. |g' x| *R vec (f(g x))) = (vec b :: real ^ 1)
  ⇔ (λx. vec (f x) :: real ^ 1) absolutely-integrable-on (g ' S) ∧
  integral (g ' S) (λx. vec (f x)) = (vec b :: real ^ 1)
using assms unfolding has-real-derivative-iff-has-vector-derivative
by (intro has-absolute-integral-change-of-variables-1 assms) auto
thus ?thesis
by (simp add: absolutely-integrable-on-1-iff integral-on-1-eq)
qed

```

```

lemma uniform-limit-set-lebesgue-integral:
  fixes  $f :: 'a \Rightarrow 'b :: euclidean-space \Rightarrow 'c :: \{banach, second-countable-topology\}$ 
  assumes set-integrable lborel  $X' g$ 
  assumes [measurable]:  $X' \in sets\ borel$ 
  assumes [measurable]:  $\bigwedge y. y \in Y \implies set-borel-measurable\ borel\ X' (f\ y)$ 
  assumes  $\bigwedge y. y \in Y \implies (\exists t \in X' \text{ in } lborel. norm (f\ y\ t) \leq g\ t)$ 
  assumes eventually  $(\lambda x. X\ x \in sets\ borel \wedge X\ x \subseteq X') F$ 
  assumes filterlim  $(\lambda x. set-lebesgue-integral\ lborel (X\ x)\ g)$ 
     $(nhds (set-lebesgue-integral\ lborel\ X'\ g)) F$ 
  shows uniform-limit  $Y$ 
     $(\lambda x\ y. set-lebesgue-integral\ lborel (X\ x)\ (f\ y))$ 
     $(\lambda y. set-lebesgue-integral\ lborel\ X'\ (f\ y)) F$ 
proof (rule uniform-limitI, goal-cases)
  case ( $1\ \varepsilon$ )
  have integrable-g: set-integrable lborel  $U g$ 
  if  $U \in sets\ borel\ U \subseteq X'$  for  $U$ 
  by (rule set-integrable-subset[OF assms(1)]) (use that in auto)
  have eventually  $(\lambda x. dist (set-lebesgue-integral\ lborel (X\ x)\ g)$ 
     $(set-lebesgue-integral\ lborel\ X'\ g) < \varepsilon) F$ 
  using  $\langle \varepsilon > 0 \rangle$  assms by (auto simp: tendsto-iff)
  from this show ?case using  $\langle eventually (\lambda-. - \wedge -) F \rangle$ 
proof eventually-elim
  case (elim x)
  hence [measurable]:  $X\ x \in sets\ borel$  and  $X\ x \subseteq X'$  by auto
  have integrable: set-integrable lborel  $U (f\ y)$ 
  if  $y \in Y\ U \in sets\ borel\ U \subseteq X'$  for  $y\ U$ 
  apply (rule set-integrable-subset)
  apply (rule set-integrable-bound[OF assms(1)])
  apply (use assms(3) that in <simp add: set-borel-measurable-def>)
  using assms(4)[OF <y ∈ Y>] apply eventually-elim apply force
  using that apply simp-all
  done
  show ?case
proof
  fix  $y$  assume  $y \in Y$ 
  have dist  $(set-lebesgue-integral\ lborel (X\ x)\ (f\ y))$ 
     $(set-lebesgue-integral\ lborel\ X'\ (f\ y)) =$ 
     $norm (set-lebesgue-integral\ lborel\ X'\ (f\ y) -$ 
     $set-lebesgue-integral\ lborel (X\ x)\ (f\ y))$ 
  by (simp add: dist-norm norm-minus-commute)
  also have set-lebesgue-integral lborel  $X' (f\ y) -$ 
     $set-lebesgue-integral\ lborel (X\ x)\ (f\ y) =$ 
     $set-lebesgue-integral\ lborel (X' - X\ x)\ (f\ y)$ 
  unfolding set-lebesgue-integral-def
  apply (subst Bochner-Integration.integral-diff [symmetric])
  unfolding set-integrable-def [symmetric]
  apply (rule integrable; (fact | simp))
  apply (rule integrable; fact)
  apply (intro Bochner-Integration.integral-cong)

```



```

    apply (use ⟨X x ⊆ X'⟩ in ⟨auto simp: indicator-def⟩)
  done
  also have norm ... ≤ (∫ t∈X'-X x. norm (f y t) ∂lborel)
    by (intro set-integral-norm-bound integrable) (fact | simp)+
  also have AE t∈X' - X x in lborel. norm (f y t) ≤ g t
    using assms(4)[OF ⟨y ∈ Y⟩] by eventually-elim auto
  with ⟨y ∈ Y⟩ have (∫ t∈X'-X x. norm (f y t) ∂lborel) ≤ (∫ t∈X'-X x. g t
∂lborel)
    by (intro set-integral-mono-AE set-integrable-norm integrable integrable-g)
  auto
  also have ... = (∫ t∈X'. g t ∂lborel) - (∫ t∈X x. g t ∂lborel)
  unfolding set-lebesgue-integral-def
  apply (subst Bochner-Integration.integral-diff [symmetric])
  unfolding set-integrable-def [symmetric]
  apply (rule integrable-g; (fact | simp))
  apply (rule integrable-g; fact)
  apply (intro Bochner-Integration.integral-cong)
  apply (use ⟨X x ⊆ X'⟩ in ⟨auto simp: indicator-def⟩)
  done
  also have ... ≤ dist (∫ t∈X x. g t ∂lborel) (∫ t∈X'. g t ∂lborel)
    by (simp add: dist-norm)
  also have ... < ε by fact
  finally show dist (set-lebesgue-integral lborel (X x) (f y))
    (set-lebesgue-integral lborel X' (f y)) < ε .

qed
qed
qed

```

**lemma** *integral-dominated-convergence-at-right*:

```

fixes s :: real ⇒ 'a ⇒ 'b::{banach, second-countable-topology} and w :: 'a ⇒ real
  and f :: 'a ⇒ 'b and M and c :: real
  assumes f ∈ borel-measurable M ∧ t. s t ∈ borel-measurable M integrable M w
  assumes lim: AE x in M. ((λi. s i x) ⟶ f x) (at-right c)
  assumes bound: ∀ F i in at-right c. AE x in M. norm (s i x) ≤ w x
  shows ((λt. integralL M (s t)) ⟶ integralL M f) (at-right c)
proof (rule tendsto-at-right-reall-sequentially)
  fix X :: nat ⇒ real assume X: filterlim X (at-right c) sequentially
  from filterlim-iff[THEN iffD1, OF this, rule-format, OF bound]
  obtain N where w: ∧n. N ≤ n ⟹ AE x in M. norm (s (X n) x) ≤ w x
  by (auto simp: eventually-sequentially)

```

**show** (λn. integral<sup>L</sup> M (s (X n))) ⟶ integral<sup>L</sup> M f

**proof** (rule LIMSEQ-offset, rule integral-dominated-convergence)

**show** AE x in M. norm (s (X (n + N)) x) ≤ w x **for** n

**by** (rule w) auto

**show** AE x in M. (λn. s (X (n + N)) x) ⟶ f x

**using** lim

**proof** eventually-elim

**fix** x **assume** ((λi. s i x) ⟶ f x) (at-right c)

**then show**  $(\lambda n. s (X (n + N)) x) \longrightarrow f x$   
**by** (*intro LIMSEQ-ignore-initial-segment filterlim-compose*[*OF - X*])  
**qed**  
**qed fact+**  
**qed**

**lemma** *integral-dominated-convergence-at-left*:

**fixes**  $s :: \text{real} \Rightarrow 'a \Rightarrow 'b :: \{\text{banach, second-countable-topology}\}$  **and**  $w :: 'a \Rightarrow \text{real}$   
**and**  $f :: 'a \Rightarrow 'b$  **and**  $M$  **and**  $c :: \text{real}$   
**assumes**  $f \in \text{borel-measurable } M \wedge t. s t \in \text{borel-measurable } M \text{ integrable } M w$   
**assumes**  $\text{lim}: AE x \text{ in } M. ((\lambda i. s i x) \longrightarrow f x) \text{ (at-left } c)$   
**assumes**  $\text{bound}: \forall_F i \text{ in at-left } c. AE x \text{ in } M. \text{norm } (s i x) \leq w x$   
**shows**  $((\lambda t. \text{integral}^L M (s t)) \longrightarrow \text{integral}^L M f) \text{ (at-left } c)$   
**proof** (*rule tendsto-at-left-realI-sequentially*)  
**fix**  $X :: \text{nat} \Rightarrow \text{real}$  **assume**  $X: \text{filterlim } X \text{ (at-left } c) \text{ sequentially}$   
**from** *filterlim-iff*[*THEN iffD1, OF this, rule-format, OF bound*]  
**obtain**  $N$  **where**  $w: \bigwedge n. N \leq n \implies AE x \text{ in } M. \text{norm } (s (X n) x) \leq w x$   
**by** (*auto simp: eventually-sequentially*)

**show**  $(\lambda n. \text{integral}^L M (s (X n))) \longrightarrow \text{integral}^L M f$

**proof** (*rule LIMSEQ-offset, rule integral-dominated-convergence*)

**show**  $AE x \text{ in } M. \text{norm } (s (X (n + N)) x) \leq w x$  **for**  $n$

**by** (*rule w*) *auto*

**show**  $AE x \text{ in } M. (\lambda n. s (X (n + N)) x) \longrightarrow f x$

**using** *lim*

**proof** *eventually-elim*

**fix**  $x$  **assume**  $((\lambda i. s i x) \longrightarrow f x) \text{ (at-left } c)$

**then show**  $(\lambda n. s (X (n + N)) x) \longrightarrow f x$

**by** (*intro LIMSEQ-ignore-initial-segment filterlim-compose*[*OF - X*])

**qed**

**qed fact+**

**qed**

**lemma** *uniform-limit-interval-integral-right*:

**fixes**  $f :: 'a \Rightarrow \text{real} \Rightarrow 'c :: \{\text{banach, second-countable-topology}\}$

**assumes** *interval-lebesgue-integrable lborel*  $a b g$

**assumes** [*measurable*]:  $\bigwedge y. y \in Y \implies \text{set-borel-measurable borel (einterval } a b)$   
 $(f y)$

**assumes**  $\bigwedge y. y \in Y \implies (AE t \in \text{einterval } a b \text{ in lborel. norm } (f y t) \leq g t)$

**assumes**  $a < b$

**shows** *uniform-limit*  $Y (\lambda b' y. \text{LBINT } t=a..b'. f y t) (\lambda y. \text{LBINT } t=a..b. f y t) \text{ (at-left } b)$

**proof** (*cases*  $Y = \{\}$ )

**case** *False*

**have** *g-nonneg*:  $AE t \in \text{einterval } a b \text{ in lborel. } g t \geq 0$

**proof**  $-$

**from**  $\langle Y \neq \{\} \rangle$  **obtain**  $y$  **where**  $y \in Y$  **by** *auto*

**from** *assms*(3)[*OF this*] **show** *?thesis*

**by** *eventually-elim (auto elim: order.trans[rotated])*

qed

**have**  $ev$ : eventually  $(\lambda b'. b' \in \{a < .. < b\})$  (at-left b)  
**using**  $\langle a < b \rangle$  **by** (intro eventually-at-leftI)  
**with**  $\langle a < b \rangle$  **have** ?thesis  $\longleftrightarrow$  uniform-limit  $Y$   $(\lambda b' y. \int t \in \text{einterval } a \text{ (min } b \text{ } b'). f y t \partial \text{lborel})$   
 $(\lambda y. \int t \in \text{einterval } a \text{ } b. f y t \partial \text{lborel})$  (at-left b)  
**by** (intro filterlim-cong arg-cong2[**where**  $f = \text{uniformly-on}$ ])  
(auto simp: interval-lebesgue-integral-def fun-eq-iff min-def  
intro!: eventually-mono[OF ev])  
**also have** ...  
**proof** (rule uniform-limit-set-lebesgue-integral[**where**  $g = g$ ], goal-cases)  
**show**  $\forall_F b'$  in at-left b.  $\text{einterval } a \text{ (min } b \text{ } b') \in \text{sets borel } \wedge$   
 $\text{einterval } a \text{ (min } b \text{ } b') \subseteq \text{einterval } a \text{ } b$   
**using**  $ev$  **by** eventually-elim (auto simp: einterval-def)  
**next**  
**show**  $(\lambda b'. \text{set-lebesgue-integral lborel (einterval } a \text{ (min } b \text{ } b')) g) \longrightarrow$   
 $\text{set-lebesgue-integral lborel (einterval } a \text{ } b) g$  (at-left b)  
**unfolding** set-lebesgue-integral-def  
**proof** (intro tendsto-at-left-erealI-sequentially integral-dominated-convergence)  
**have**  $*$ : set-borel-measurable borel (einterval a b) g  
**using**  $\text{assms}(1)$  less-imp-le[OF  $\langle a < b \rangle$ ]  
**by** (simp add: interval-lebesgue-integrable-def set-integrable-def set-borel-measurable-def)  
**show**  $(\lambda x. \text{indicat-real (einterval } a \text{ } b) x *_R g x) \in \text{borel-measurable lborel}$   
**using**  $*$  **by** (simp add: set-borel-measurable-def)  
**fix**  $X :: \text{nat} \Rightarrow \text{ereal}$  **and**  $n :: \text{nat}$   
**have** set-borel-measurable borel (einterval a (min b (X n))) g  
**by** (rule set-borel-measurable-subset[OF  $*$ ]) (auto simp: einterval-def)  
**thus**  $(\lambda x. \text{indicat-real (einterval } a \text{ (min } b \text{ (X } n))} x *_R g x) \in \text{borel-measurable}$   
 $\text{lborel}$   
**by** (simp add: set-borel-measurable-def)  
**next**  
**fix**  $X :: \text{nat} \Rightarrow \text{ereal}$   
**assume**  $X$ : filterlim  $X$  (at-left b) sequentially  
**show**  $AE x$  in lborel.  $(\lambda n. \text{indicat-real (einterval } a \text{ (min } b \text{ (X } n))} x *_R g x)$   
 $\longrightarrow \text{indicat-real (einterval } a \text{ } b) x *_R g x$   
**proof** (rule AE-I2)  
**fix**  $x :: \text{real}$   
**have**  $(\lambda t. \text{indicator (einterval } a \text{ (min } b \text{ (X } t))} x :: \text{real}) \longrightarrow$   
 $\text{indicator (einterval } a \text{ } b) x$   
**proof** (cases  $x \in \text{einterval } a \text{ } b$ )  
**case** False  
**hence**  $x \notin \text{einterval } a \text{ (min } b \text{ (X } t))$  **for**  $t$  **by** (auto simp: einterval-def)  
**with** False **show** ?thesis **by** (simp add: indicator-def)  
**next**  
**case** True  
**with**  $\langle a < b \rangle$  **have** eventually  $(\lambda t. t \in \{\max a x < .. < b\})$  (at-left b)  
**by** (intro eventually-at-leftI[of  $\text{ereal } x$ ]) (auto simp: einterval-def min-def)  
**from** this **and**  $X$  **have** eventually  $(\lambda t. X t \in \{\max a x < .. < b\})$  sequentially

by (rule eventually-compose-filterlim)  
 hence eventually ( $\lambda t. \text{indicator } (einterval\ a\ (\min\ b\ (X\ t)))\ x = (1 :: \text{real})$ )  
 sequentially  
 by eventually-elim (use True in ⟨auto simp: indicator-def einterval-def⟩)  
 from tendsto-eventually[OF this] and True show ?thesis  
 by (simp add: indicator-def)  
 qed  
 thus ( $\lambda n. \text{indicat-real } (einterval\ a\ (\min\ b\ (X\ n)))\ x *_R\ g\ x$ )  
 —————→  $\text{indicat-real } (einterval\ a\ b)\ x *_R\ g\ x$  by (intro tendsto-intros)  
 qed  
 next  
 fix  $X :: \text{nat} \Rightarrow \text{ereal}$  and  $n :: \text{nat}$   
 show  $AE\ x\ \text{in}\ \text{lborel}. \text{norm } (\text{indicator } (einterval\ a\ (\min\ b\ (X\ n)))\ x *_R\ g\ x)$   
 $\leq$   
 $\text{indicator } (einterval\ a\ b)\ x *_R\ g\ x$   
 using  $g\text{-nonneg}$  by eventually-elim (auto simp: indicator-def einterval-def)  
 qed (use assms less-imp-le[OF ⟨ $a < b$ ⟩] in  
 ⟨auto simp: interval-lebesgue-integrable-def set-integrable-def⟩)  
 qed (use assms in ⟨auto simp: interval-lebesgue-integrable-def⟩)  
 finally show ?thesis .  
 qed auto  
  
**lemma** *uniform-limit-interval-integral-left*:  
 fixes  $f :: 'a \Rightarrow \text{real} \Rightarrow 'c :: \{\text{banach}, \text{second-countable-topology}\}$   
 assumes *interval-lebesgue-integrable*  $\text{lborel}\ a\ b\ g$   
 assumes [*measurable*]:  $\bigwedge y. y \in Y \implies \text{set-borel-measurable borel } (einterval\ a\ b)$   
 ( $f\ y$ )  
 assumes  $\bigwedge y. y \in Y \implies (AE\ t \in einterval\ a\ b\ \text{in}\ \text{lborel}. \text{norm } (f\ y\ t) \leq g\ t)$   
 assumes  $a < b$   
 shows *uniform-limit*  $Y\ (\lambda a'\ y. \text{LBINT}\ t=a'..b. f\ y\ t)\ (\lambda y. \text{LBINT}\ t=a..b. f\ y\ t)$  (*at-right*  $a$ )  
**proof** (cases  $Y = \{\}$ )  
 case False  
 have  $g\text{-nonneg}$ :  $AE\ t \in einterval\ a\ b\ \text{in}\ \text{lborel}. g\ t \geq 0$   
**proof** –  
 from  $\langle Y \neq \{\} \rangle$  obtain  $y$  where  $y \in Y$  by auto  
 from *assms*( $\beta$ )[OF this] show ?thesis  
 by eventually-elim (auto elim: *order.trans*[rotated])  
 qed  
  
 have *ev*: eventually ( $\lambda b'. b' \in \{a <..<b\}$ ) (*at-right*  $a$ )  
 using  $\langle a < b \rangle$  by (intro eventually-at-rightI)  
 with  $\langle a < b \rangle$  have ?thesis  $\iff$  *uniform-limit*  $Y\ (\lambda a'\ y. \int t \in einterval\ (\max\ a\ a')\ b. f\ y\ t\ \partial \text{lborel})$   
 ( $\lambda y. \int t \in einterval\ a\ b. f\ y\ t\ \partial \text{lborel}$ ) (*at-right*  $a$ )  
 by (intro filterlim-cong arg-cong2[**where**  $f = \text{uniformly-on}$ ])  
 (auto simp: interval-lebesgue-integral-def fun-eq-iff max-def  
 intro!: eventually-mono[OF *ev*])  
 also have ...

```

proof (rule uniform-limit-set-lebesgue-integral[where  $g = g$ ], goal-cases)
  show  $\forall_F a'$  in at-right  $a$ .  $einterval (max a a') b \in sets borel \wedge$ 
     $einterval (max a a') b \subseteq einterval a b$ 
  using  $ev$  by eventually-elim (auto simp: einterval-def)
next
  show  $((\lambda a'. set-lebesgue-integral lborel (einterval (max a a') b) g) \longrightarrow$ 
     $set-lebesgue-integral lborel (einterval a b) g)$  (at-right  $a$ )
  unfolding set-lebesgue-integral-def
proof (intro tendsto-at-right-erealI-sequentially integral-dominated-convergence)
  have  $*$ : set-borel-measurable borel (einterval  $a b$ )  $g$ 
    using  $assms(1)$  less-imp-le[OF  $\langle a < b \rangle$ ]
  by (simp add: interval-lebesgue-integrable-def set-integrable-def set-borel-measurable-def)
  show  $(\lambda x. indicat-real (einterval a b) x *_R g x) \in borel-measurable lborel$ 
    using  $*$  by (simp add: set-borel-measurable-def)
  fix  $X :: nat \Rightarrow ereal$  and  $n :: nat$ 
  have set-borel-measurable borel (einterval (max  $a (X n)$ )  $b$ )  $g$ 
    by (rule set-borel-measurable-subset[OF  $*$ ]) (auto simp: einterval-def)
  thus  $(\lambda x. indicat-real (einterval (max a (X n)) b) x *_R g x) \in borel-measurable$ 
lborel
    by (simp add: set-borel-measurable-def)
next
  fix  $X :: nat \Rightarrow ereal$ 
  assume  $X$ : filterlim  $X$  (at-right  $a$ ) sequentially
  show  $AE x$  in  $lborel$ .  $(\lambda n. indicat-real (einterval (max a (X n)) b) x *_R g x)$ 
     $\longrightarrow indicat-real (einterval a b) x *_R g x$ 
proof (rule AE-I2)
  fix  $x :: real$ 
  have  $(\lambda t. indicator (einterval (max a (X t)) b) x :: real) \longrightarrow$ 
     $indicator (einterval a b) x$ 
  proof (cases  $x \in einterval a b$ )
  case  $False$ 
  hence  $x \notin einterval (max a (X t)) b$  for  $t$  by (auto simp: einterval-def)
  with  $False$  show ?thesis by (simp add: indicator-def)
  next
  case  $True$ 
  with  $\langle a < b \rangle$  have eventually  $(\lambda t. t \in \{a <..<x\})$  (at-right  $a$ )
    by (intro eventually-at-rightI[of - ereal  $x$ ]) (auto simp: einterval-def
min-def)
  from this and  $X$  have eventually  $(\lambda t. X t \in \{a <..<x\})$  sequentially
    by (rule eventually-compose-filterlim)
  hence eventually  $(\lambda t. indicator (einterval (max a (X t)) b) x = (1 :: real))$ 
sequentially
    by eventually-elim (use  $True$  in  $\langle auto simp: indicator-def einterval-def \rangle$ )
  from tendsto-eventually[OF this] and  $True$  show ?thesis
    by (simp add: indicator-def)
  qed
  thus  $(\lambda n. indicat-real (einterval (max a (X n)) b) x *_R g x)$ 
     $\longrightarrow indicat-real (einterval a b) x *_R g x$  by (intro tendsto-intros)
qed

```

```

next
  fix X :: nat ⇒ ereal and n :: nat
  show AE x in lborel. norm (indicator (einterval (max a (X n)) b) x *R g x)
≤
  indicator (einterval a b) x *R g x
  using g-nonneg by eventually-elim (auto simp: indicator-def einterval-def)
  qed (use assms less-imp-le[OF ‹a < b›] in
    ‹auto simp: interval-lebesgue-integrable-def set-integrable-def›)
  qed (use assms in ‹auto simp: interval-lebesgue-integrable-def›)
  finally show ?thesis .
qed auto

lemma uniform-limit-interval-integral-sequentially:
  fixes f :: 'a ⇒ real ⇒ 'c :: {banach, second-countable-topology}
  assumes interval-lebesgue-integrable lborel a b g
  assumes [measurable]: ∧y. y ∈ Y ⇒ set-borel-measurable borel (einterval a b)
  (f y)
  assumes ∧y. y ∈ Y ⇒ (AE t∈einterval a b in lborel. norm (f y t) ≤ g t)
  assumes a': filterlim a' (at-right a) sequentially
  assumes b': filterlim b' (at-left b) sequentially
  assumes a < b
  shows uniform-limit Y (λn y. LBINT t=a' n..b' n. f y t)
    (λy. LBINT t=a..b. f y t) sequentially
proof (cases Y = {})
  case False
  have g-nonneg: AE t∈einterval a b in lborel. g t ≥ 0
  proof -
    from ‹Y ≠ {}› obtain y where y ∈ Y by auto
    from assms(3)[OF this] show ?thesis
    by eventually-elim (auto elim: order.trans[rotated])
  qed
  have ev: eventually (λn. a < a' n ∧ a' n < b' n ∧ b' n < b) sequentially
  proof -
    from ereal-dense2[OF ‹a < b›] obtain t where t: a < ereal t ereal t < b by
    blast
    from t have eventually (λn. a' n ∈ {a<..

```

```

      (auto simp: interval-lebesgue-integral-def fun-eq-iff min-def max-def
        intro!: eventually-mono[OF ev])
    also have ...
  proof (rule uniform-limit-set-lebesgue-integral[where g = g], goal-cases)
    show  $\forall_F n$  in sequentially.  $einterval (max a (a' n)) (min b (b' n)) \in sets borel$ 
  ^
       $einterval (max a (a' n)) (min b (b' n)) \subseteq einterval a b$ 
    using ev by eventually-elim (auto simp: einterval-def)
  next
    show (( $\lambda n$ .  $set-lebesgue-integral lborel (einterval (max a (a' n)) (min b (b' n)))$ )
  g)  $\longrightarrow$ 
       $set-lebesgue-integral lborel (einterval a b) g$  sequentially
    unfolding set-lebesgue-integral-def
  proof (intro integral-dominated-convergence)
    have *:  $set-borel-measurable borel (einterval a b) g$ 
      using  $assms(1)$  less-imp-le[OF <a < b>]
    by (simp add: interval-lebesgue-integrable-def set-integrable-def set-borel-measurable-def)
    show ( $\lambda x$ .  $indicat-real (einterval a b) x *_R g x$ )  $\in borel-measurable lborel$ 
      using * by (simp add: set-borel-measurable-def)
    fix n :: nat
    have  $set-borel-measurable borel (einterval (max a (a' n)) (min b (b' n))) g$ 
      by (rule set-borel-measurable-subset[OF *]) (auto simp: einterval-def)
    thus ( $\lambda x$ .  $indicat-real (einterval (max a (a' n)) (min b (b' n))) x *_R g x$ )  $\in$ 
  borel-measurable lborel
      by (simp add: set-borel-measurable-def)
    next
    show  $AE x$  in lborel. ( $\lambda n$ .  $indicat-real (einterval (max a (a' n)) (min b (b' n))) x *_R g x$ )
   $\longrightarrow$   $indicat-real (einterval a b) x *_R g x$ 
  proof (rule AE-I2)
    fix x :: real
    have ( $\lambda t$ .  $indicator (einterval (max a (a' t)) (min b (b' t))) x :: real$ )  $\longrightarrow$ 
       $indicator (einterval a b) x$ 
    proof (cases  $x \in einterval a b$ )
      case False
      hence  $x \notin einterval (max a (a' t)) (min b (b' t))$  for t
        by (auto simp: einterval-def)
      with False show ?thesis by (simp add: indicator-def)
    next
      case True
      with <a < b> have eventually ( $\lambda t$ .  $t \in \{a <..<x\}$ ) (at-right a)
        by (intro eventually-at-rightI[of - ereal x]) (auto simp: einterval-def
  min-def)

      have eventually ( $\lambda n$ .  $x \in \{a' n <..<b' n\}$ ) sequentially
    proof -
      have eventually ( $\lambda n$ .  $a' n \in \{a <..<x\}$ ) sequentially using True
        by (intro eventually-compose-filterlim[OF - a'] eventually-at-rightI[of -
  ereal x])

```

```

      (auto simp: einterval-def)
      moreover have eventually ( $\lambda n. b' n \in \{x <..<b\}$ ) sequentially using
True
      by (intro eventually-compose-filterlim[OF - b'] eventually-at-leftI[of
ereal x])
      (auto simp: einterval-def)
      ultimately show eventually ( $\lambda n. x \in \{a' n <..<b' n\}$ ) sequentially
      by eventually-elim auto
    qed
    hence eventually ( $\lambda t. \text{indicator } (einterval (max a (a' t)) (min b (b' t))) x$ 
= (1 :: real)) sequentially
      by eventually-elim (use True in <auto simp: indicator-def einterval-def>)
      from tendsto-eventually[OF this] and True show ?thesis
      by (simp add: indicator-def)
    qed
    thus ( $\lambda n. \text{indicat-real } (einterval (max a (a' n)) (min b (b' n))) x *_R g x$ 
 $\longrightarrow \text{indicat-real } (einterval a b) x *_R g x$ ) by (intro tendsto-intros)
  qed
next
  fix X :: nat  $\Rightarrow$  ereal and n :: nat
  show AE x in lborel. norm (indicator (einterval (max a (a' n)) (min b (b'
n))) x *_R g x)  $\leq$ 
    indicator (einterval a b) x *_R g x
    using g-nonneg by eventually-elim (auto simp: indicator-def einterval-def)
  qed (use assms less-imp-le[OF <a < b>] in
    <auto simp: interval-lebesgue-integrable-def set-integrable-def>)
  qed (use assms in <auto simp: interval-lebesgue-integrable-def>)
  finally show ?thesis .
qed auto

lemma interval-lebesgue-integrable-combine:
  assumes interval-lebesgue-integrable lborel A B f
  assumes interval-lebesgue-integrable lborel B C f
  assumes set-borel-measurable borel (einterval A C) f
  assumes A  $\leq$  B B  $\leq$  C
  shows interval-lebesgue-integrable lborel A C f
proof -
  have meas: set-borel-measurable borel (einterval A B  $\cup$  einterval B C) f
  by (rule set-borel-measurable-subset[OF assms(?)]) (use assms in <auto simp:
einterval-def>)
  have set-integrable lborel (einterval A B  $\cup$  einterval B C) f
  using assms by (intro set-integrable-Un) (auto simp: interval-lebesgue-integrable-def)
  also have ?this  $\iff$  set-integrable lborel (einterval A C) f
  proof (cases B  $\in$   $\{\infty, -\infty\}$ )
  case True
  with assms have einterval A B  $\cup$  einterval B C = einterval A C
  by (auto simp: einterval-def)
  thus ?thesis by simp
  next

```



**case** *False*  
**then obtain**  $B'$  **where**  $[simp]: B = ereal B'$   
**by**  $(cases B)$  *auto*  
**have**  $indicator (einterval A C) x = (indicator (einterval A B \cup einterval B C)$   
 $x :: real)$   
**if**  $x \neq B'$  **for**  $x$  **using**  $assms(4,5)$  *that*  
**by**  $(cases A; cases C)$   $(auto simp: einterval-def indicator-def)$   
**hence**  $\{x \in space lborel. indicat-real (einterval A B \cup einterval B C) x *_R f x$   
 $\neq$   
 $indicat-real (einterval A C) x *_R f x\} \subseteq \{B'\}$  **by force**  
**thus** *?thesis* **unfolding** *set-integrable-def* **using** *meas assms*  
**by**  $(intro integrable-cong-AE AE-I[of - - \{B'\}])$   
 $(simp-all add: set-borel-measurable-def)$   
**qed**  
**also have**  $\dots \longleftrightarrow ?thesis$   
**using**  $order.trans[OF assms(4,5)]$  **by**  $(simp add: interval-lebesgue-integrable-def)$   
**finally show** *?thesis* .  
**qed**

**lemma** *interval-lebesgue-integrable-bigo-right:*

**fixes**  $A B :: real$   
**fixes**  $f :: real \Rightarrow real$   
**assumes**  $f \in O[at-left B](g)$   
**assumes** *cont: continuous-on*  $\{A..<B\}$   $f$   
**assumes** *meas: set-borel-measurable borel*  $\{A<..<<B\}$   $f$   
**assumes** *interval-lebesgue-integrable lborel*  $A B g$   
**assumes**  $A < B$   
**shows** *interval-lebesgue-integrable lborel*  $A B f$   
**proof** –  
**from**  $assms(1)$  **obtain**  $c$  **where**  $c: c > 0$  *eventually*  $(\lambda x. norm (f x) \leq c * norm$   
 $(g x)) (at-left B)$   
**by**  $(elim landau-o.bigE)$   
**then obtain**  $B'$  **where**  $B': B' < B \wedge x. x \in \{B'<..<<B\} \implies norm (f x) \leq c * norm$   
 $(g x)$   
**using**  $\langle A < B \rangle$  **by**  $(auto simp: Topological-Spaces.eventually-at-left[of A])$

**show** *?thesis*

**proof**  $(rule interval-lebesgue-integrable-combine)$

**show** *interval-lebesgue-integrable lborel*  $A (max A B') f$

**using**  $B'$  *assms*

**by**  $(intro interval-integrable-continuous-on continuous-on-subset[OF cont])$

*auto*

**show** *set-borel-measurable borel*  $(einterval (ereal A) (ereal B)) f$

**using** *assms* **by** *simp*

**have** *meas'*: *set-borel-measurable borel*  $\{max A B'<..<<B\}$   $f$

**by**  $(rule set-borel-measurable-subset[OF meas])$  *auto*

**have** *set-integrable lborel*  $\{max A B'<..<<B\}$   $f$

**proof**  $(rule set-integrable-bound[OF - - AE-I2[OF impI]])$

**have** *set-integrable lborel*  $\{A<..<<B\}$   $(\lambda x. c * g x)$

```

    using assms by (simp add: interval-lebesgue-integrable-def)
  thus set-integrable lborel {max A B' <..<B} (λx. c * g x)
    by (rule set-integrable-subset) auto
next
fix x assume x ∈ {max A B' <..<B}
hence norm (f x) ≤ c * norm (g x)
  by (intro B') auto
also have ... ≤ norm (c * g x)
  unfolding norm-mult by (intro mult-right-mono) auto
finally show norm (f x) ≤ norm (c * g x) .
qed (use meas' in ⟨simp-all add: set-borel-measurable-def⟩)
thus interval-lebesgue-integrable lborel (ereal (max A B')) (ereal B) f
  unfolding interval-lebesgue-integrable-def einterval-eq-Icc using ⟨B' < B⟩
assms by simp
qed (use B' assms in auto)
qed

lemma interval-lebesgue-integrable-bigo-left:
  fixes A B :: real
  fixes f :: real ⇒ real
  assumes f ∈ O[at-right A](g)
  assumes cont: continuous-on {A<..B} f
  assumes meas: set-borel-measurable borel {A<..<B} f
  assumes interval-lebesgue-integrable lborel A B g
  assumes A < B
  shows interval-lebesgue-integrable lborel A B f
proof -
  from assms(1) obtain c where c: c > 0 eventually (λx. norm (f x) ≤ c * norm
(g x)) (at-right A)
  by (elim landau-o.bigE)
  then obtain A' where A': A' > A ∧ x. x ∈ {A<..<A'} ⇒ norm (f x) ≤ c *
norm (g x)
  using ⟨A < B⟩ by (auto simp: Topological-Spaces.eventually-at-right[of A])

  show ?thesis
proof (rule interval-lebesgue-integrable-combine)
  show interval-lebesgue-integrable lborel (min B A') B f
  using A' assms
  by (intro interval-integrable-continuous-on continuous-on-subset[OF cont])
auto
  show set-borel-measurable borel (einterval (ereal A) (ereal B)) f
  using assms by simp
  have meas': set-borel-measurable borel {A<..<min B A'} f
  by (rule set-borel-measurable-subset[OF meas]) auto
  have set-integrable lborel {A<..<min B A'} f
  proof (rule set-integrable-bound[OF - - AE-I2[OF impI]])
  have set-integrable lborel {A<..<B} (λx. c * g x)
  using assms by (simp add: interval-lebesgue-integrable-def)
  thus set-integrable lborel {A<..<min B A'} (λx. c * g x)

```

```

    by (rule set-integrable-subset) auto
next
fix x assume x ∈ {A <..< min B A'}
hence norm (f x) ≤ c * norm (g x)
  by (intro A') auto
also have ... ≤ norm (c * g x)
  unfolding norm-mult by (intro mult-right-mono) auto
finally show norm (f x) ≤ norm (c * g x) .
qed (use meas' in ⟨simp-all add: set-borel-measurable-def⟩)
thus interval-lebesgue-integrable lborel (ereal A) (ereal (min B A')) f
  unfolding interval-lebesgue-integrable-def einterval-eq-Icc using ⟨A' > A⟩
assms by simp
qed (use A' assms in auto)
qed

```

## 1.4 Other material

**lemma** *summable-comparison-test-bigo*:

```

fixes f :: nat ⇒ real
assumes summable (λn. norm (g n)) f ∈ O(g)
shows summable f
proof -
  from ⟨f ∈ O(g)⟩ obtain C where C: eventually (λx. norm (f x) ≤ C * 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** *fps-expansion-cong*:

```

assumes eventually (λx. g x = h x) (nhds x)
shows fps-expansion g x = fps-expansion h x
proof -
  have (deriv ~ n) g x = (deriv ~ n) h x for n
  by (intro higher-deriv-cong-ev assms refl)
  thus ?thesis by (simp add: fps-expansion-def)
qed

```

**lemma** *fps-expansion-eq-zero-iff*:

```

assumes g holomorphic-on ball z r r > 0
shows fps-expansion g z = 0 ⟷ (∀ z ∈ ball z r. g z = 0)
proof
  assume *: ∀ z ∈ ball z r. g z = 0
  have eventually (λw. w ∈ ball z r) (nhds z)
  using assms by (intro eventually-nhds-in-open) auto
  hence eventually (λz. g z = 0) (nhds z)
  by eventually-elim (use * in auto)
  hence fps-expansion g z = fps-expansion (λ-. 0) z
  by (intro fps-expansion-cong)

```

```

thus fps-expansion  $g z = 0$ 
  by (simp add: fps-expansion-def fps-zero-def)
next
  assume *: fps-expansion  $g z = 0$ 
  have  $g w = 0$  if  $w \in \text{ball } z r$  for  $w$ 
    by (rule holomorphic-fun-eq-0-on-ball[OF assms(1) that])
      (use * in <auto simp: fps-expansion-def fps-eq-iff>)
  thus  $\forall w \in \text{ball } z r. g w = 0$  by blast
qed

lemma fds-nth-higher-deriv:
  fds-nth  $((\text{fds-deriv } \hat{\sim} k) F) = (\lambda n. (-1) \hat{\sim} k * \text{of-real } (\ln n) \hat{\sim} k * \text{fds-nth } F n)$ 
  by (induction k) (auto simp: fds-nth-deriv fun-eq-iff simp flip: scaleR-conv-of-real)

lemma binomial-n-n-minus-one [simp]:  $n > 0 \implies n \text{ choose } (n - \text{Suc } 0) = n$ 
  by (cases n) auto

lemma has-field-derivative-complex-powr-right:
   $w \neq 0 \implies ((\lambda z. w \text{ powr } z) \text{ has-field-derivative } \text{Ln } w * w \text{ powr } z)$  (at z within A)
  by (rule DERIV-subset, rule has-field-derivative-powr-right) auto

lemmas has-field-derivative-complex-powr-right' =
  has-field-derivative-complex-powr-right[THEN DERIV-chain2]

end

```

## 2 The Hurwitz and Riemann $\zeta$ functions

**theory** *Zeta-Function*

**imports**

*Euler-MacLaurin.Euler-MacLaurin*  
*Bernoulli.Bernoulli-Zeta*  
*Dirichlet-Series.Dirichlet-Series-Analysis*  
*Winding-Number-Eval.Winding-Number-Eval*  
*HOL-Real-Asymp.Real-Asymp*  
*Zeta-Library*  
*Pure-ex.Guess*

**begin**

### 2.1 Preliminary facts

**lemma** *powr-add-minus-powr-asymptotics*:

**fixes**  $a z :: \text{complex}$

**shows**  $((\lambda z. ((1 + z) \text{ powr } a - 1) / z) \longrightarrow a)$  (*at 0*)

**proof** (*rule Lim-transform-eventually*)

**have** *eventually*  $(\lambda z :: \text{complex}. z \in \text{ball } 0 1 - \{0\})$  (*at 0*)

**using** *eventually-at-ball'[of 1 0::complex UNIV]* **by** (*simp add: dist-norm*)

**thus** *eventually*  $(\lambda z. (\sum n. (a \text{ gchoose } (\text{Suc } n)) * z \hat{\sim} n) = ((1 + z) \text{ powr } a - 1) / z)$  (*at 0*)

**proof** *eventually-elim*  
**case** (*elim z*)  
**hence**  $(\lambda n. (a \text{ gchoose } n) * z ^ n) \text{ sums } (1 + z) \text{ powr } a$   
**by** (*intro gen-binomial-complex*) *auto*  
**hence**  $(\lambda n. (a \text{ gchoose } (\text{Suc } n)) * z ^ (\text{Suc } n)) \text{ sums } ((1 + z) \text{ powr } a - 1)$   
**by** (*subst sums-Suc-iff*) *simp-all*  
**also have**  $(\lambda n. (a \text{ gchoose } (\text{Suc } n)) * z ^ (\text{Suc } n)) = (\lambda n. z * ((a \text{ gchoose } (\text{Suc } n)) * z ^ n))$   
**by** (*simp add: algebra-simps*)  
**finally have**  $(\lambda n. (a \text{ gchoose } (\text{Suc } n)) * z ^ n) \text{ sums } (((1 + z) \text{ powr } a - 1) / z)$   
**by** (*rule sums-mult-D*) (*use elim in auto*)  
**thus** *?case* **by** (*simp add: sums-iff*)  
**qed**  
**next**  
**have** *conv-radius*  $(\lambda n. a \text{ gchoose } (n + 1)) = \text{conv-radius } (\lambda n. a \text{ gchoose } n)$   
**using** *conv-radius-shift*[*of*  $\lambda n. a \text{ gchoose } n$  1] **by** *simp*  
**hence** *continuous-on*  $(\text{cball } 0 (1/2)) (\lambda z. \sum n. (a \text{ gchoose } (\text{Suc } n)) * (z - 0) ^ n)$   
**using** *conv-radius-gchoose*[*of*  $a$ ] **by** (*intro powser-continuous-suminf*) (*simp-all*)  
**hence** *isCont*  $(\lambda z. \sum n. (a \text{ gchoose } (\text{Suc } n)) * z ^ n) 0$   
**by** (*auto intro: continuous-on-interior*)  
**thus**  $(\lambda z. \sum n. (a \text{ gchoose } (\text{Suc } n)) * z ^ n) - 0 \rightarrow a$   
**by** (*auto simp: isCont-def*)  
**qed**

**lemma** *complex-powr-add-minus-powr-asymptotics*:

**fixes**  $s :: \text{complex}$   
**assumes**  $a: a > 0$  **and**  $s: \text{Re } s < 1$   
**shows** *filterlim*  $(\lambda x. \text{of-real } (x + a) \text{ powr } s - \text{of-real } x \text{ powr } s) (\text{nhds } 0) \text{ at-top}$   
**proof** (*rule Lim-transform-eventually*)  
**show** *eventually*  $(\lambda x. ((1 + \text{of-real } (a / x)) \text{ powr } s - 1) / \text{of-real } (a / x) * \text{of-real } x \text{ powr } (s - 1) * a = \text{of-real } (x + a) \text{ powr } s - \text{of-real } x \text{ powr } s) \text{ at-top}$   
**(is eventually**  $(\lambda x. ?f x / ?g x * ?h x * - = -) -$  **using** *eventually-gt-at-top*[*of*  $a$ ]  
**proof** *eventually-elim*

**case** (*elim x*)  
**have**  $?f x / ?g x * ?h x * a = ?f x * (a * ?h x / ?g x)$  **by** *simp*  
**also have**  $a * ?h x / ?g x = \text{of-real } x \text{ powr } s$   
**using** *elim a* **by** (*simp add: powr-diff*)  
**also have**  $?f x * \dots = \text{of-real } (x + a) \text{ powr } s - \text{of-real } x \text{ powr } s$   
**using** *a elim* **by** (*simp add: algebra-simps powr-times-real [symmetric]*)  
**finally show** *?case* .  
**qed**

**have** *filterlim*  $(\lambda x. \text{complex-of-real } (a / x)) (\text{nhds } (\text{complex-of-real } 0)) \text{ at-top}$   
**by** (*intro tendsto-of-real real-tendsto-divide-at-top*[*OF* *tendsto-const*] *filterlim-ident*)  
**hence** *filterlim*  $(\lambda x. \text{complex-of-real } (a / x)) (\text{at } 0) \text{ at-top}$

**using**  $a$  **by** (*intro filterlim-atI*) *auto*  
**hence**  $((\lambda x. ?f x / ?g x * ?h x * a) \longrightarrow s * 0 * a)$  *at-top* **using**  $s$   
**by** (*intro tendsto-mult filterlim-compose* [*OF powr-add-minus-powr-asymptotics*]  
*tendsto-const tendsto-neg-powr-complex-of-real filterlim-ident*) *auto*  
**thus**  $((\lambda x. ?f x / ?g x * ?h x * a) \longrightarrow 0)$  *at-top* **by** *simp*  
**qed**

**lemma** *summable-zeta*:

**assumes**  $Re\ s > 1$   
**shows** *summable*  $(\lambda n. \text{of-nat } (Suc\ n)\ \text{powr } -s)$   
**proof** –  
**have** *summable*  $(\lambda n. \exp(\text{complex-of-real } (\ln(\text{real } (Suc\ n)))) * -s)$  (**is** *summable* *?f*)  
**by** (*subst summable-Suc-iff, rule summable-complex-powr-iff*) (*use assms in auto*)  
**also have** *?f* =  $(\lambda n. \text{of-nat } (Suc\ n)\ \text{powr } -s)$   
**by** (*simp add: powr-def algebra-simps del: of-nat-Suc*)  
**finally show** *?thesis* .  
**qed**

**lemma** *summable-zeta-real*:

**assumes**  $x > 1$   
**shows** *summable*  $(\lambda n. \text{real } (Suc\ n)\ \text{powr } -x)$   
**proof** –  
**have** *summable*  $(\lambda n. \text{of-nat } (Suc\ n)\ \text{powr } -\text{complex-of-real } x)$   
**using** *assms* **by** (*intro summable-zeta*) *simp-all*  
**also have**  $(\lambda n. \text{of-nat } (Suc\ n)\ \text{powr } -\text{complex-of-real } x) = (\lambda n. \text{of-real } (\text{real } (Suc\ n)\ \text{powr } -x))$   
**by** (*subst powr-Reals-eq*) *simp-all*  
**finally show** *?thesis*  
**by** (*subst (asm) summable-complex-of-real*)  
**qed**

**lemma** *summable-hurwitz-zeta*:

**assumes**  $Re\ s > 1\ a > 0$   
**shows** *summable*  $(\lambda n. (\text{of-nat } n + \text{of-real } a)\ \text{powr } -s)$   
**proof** –  
**have** *summable*  $(\lambda n. (\text{of-nat } (Suc\ n) + \text{of-real } a)\ \text{powr } -s)$   
**proof** (*rule summable-comparison-test'* [*OF summable-zeta-real*] [*OF assms(1)*])  
)
**fix**  $n :: \text{nat}$   
**have** *norm*  $((\text{of-nat } (Suc\ n) + \text{of-real } a)\ \text{powr } -s) = (\text{real } (Suc\ n) + a)\ \text{powr } -Re\ s$   
**(is**  $?N = -$ ) **using** *assms* **by** (*simp add: norm-powr-real-powr*)  
**also have**  $\dots \leq \text{real } (Suc\ n)\ \text{powr } -Re\ s$   
**using** *assms* **by** (*intro powr-mono2'*) *auto*  
**finally show**  $?N \leq \dots$  .  
**qed**

**thus** *?thesis* **by** (*subst (asm) summable-Suc-iff*)  
**qed**

**lemma** *summable-hurwitz-zeta-real*:

**assumes**  $x > 1$   $a > 0$

**shows** *summable* ( $\lambda n. (\text{real } n + a) \text{ powr } -x$ )

**proof** –

**have** *summable* ( $\lambda n. (\text{of-nat } n + \text{of-real } a) \text{ powr } -\text{complex-of-real } x$ )

**using** *assms* **by** (*intro summable-hurwitz-zeta simp-all*)

**also have** ( $\lambda n. (\text{of-nat } n + \text{of-real } a) \text{ powr } -\text{complex-of-real } x$ ) =  
 $(\lambda n. \text{of-real } ((\text{real } n + a) \text{ powr } -x))$

**using** *assms* **by** (*subst powr-Reals-eq simp-all*)

**finally show** *?thesis*

**by** (*subst (asm) summable-complex-of-real*)

**qed**

## 2.2 Definitions

We use the Euler–MacLaurin summation formula to express  $\zeta(s, a) - \frac{a^{1-s}}{s-1}$  as a polynomial plus some remainder term, which is an integral over a function of order  $O(-1 - 2n - \Re(s))$ . It is then clear that this integral converges uniformly to an analytic function in  $s$  for all  $s$  with  $\Re(s) > -2n$ .

**definition** *pre-zeta-aux* :: *nat*  $\Rightarrow$  *real*  $\Rightarrow$  *complex*  $\Rightarrow$  *complex* **where**

*pre-zeta-aux*  $N$   $a$   $s = a \text{ powr } -s / 2 +$

$(\sum_{i=1..N}. (\text{bernoulli } (2 * i) / \text{fact } (2 * i)) *_R (\text{pochhammer } s (2*i - 1) *$   
 $\text{of-real } a \text{ powr } (-s - \text{of-nat } (2*i - 1)))) +$

*EM-remainder* (*Suc* ( $2*N$ ))

$(\lambda x. -(\text{pochhammer } s (\text{Suc } (2*N)) * \text{of-real } (x + a) \text{ powr } (-1 - 2*N - s))) 0$

By iterating the above construction long enough, we can extend this to the entire complex plane.

**definition** *pre-zeta* :: *real*  $\Rightarrow$  *complex*  $\Rightarrow$  *complex* **where**

*pre-zeta*  $a$   $s = \text{pre-zeta-aux } (\text{nat } (1 - \lceil \text{Re } s / 2 \rceil)) a s$

We can then obtain the Hurwitz  $\zeta$  function by adding back the pole at 1. Note that it is not necessary to trust that this somewhat complicated definition is, in fact, the correct one, since we will later show that this Hurwitz zeta function fulfils

$$\zeta(s, a) = \sum_{n=0}^{\infty} \frac{1}{(n+a)^s}$$

and is analytic on  $\mathbb{C} \setminus \{1\}$ , which uniquely defines the function due to analytic continuation. It is therefore obvious that any alternative definition that is analytic on  $\mathbb{C} \setminus \{1\}$  and satisfies the above equation must be equal to our Hurwitz  $\zeta$  function.

**definition** *hurwitz-zeta* :: *real*  $\Rightarrow$  *complex*  $\Rightarrow$  *complex* **where**

*hurwitz-zeta* *a s* = (if *s* = 1 then 0 else *pre-zeta a s* + *of-real a powr (1 - s) / (s - 1)*)

The Riemann  $\zeta$  function is simply the Hurwitz  $\zeta$  function with  $a = 1$ .

**definition** *zeta* :: *complex*  $\Rightarrow$  *complex* **where**

*zeta* = *hurwitz-zeta 1*

We define the  $\zeta$  functions as 0 at their poles. To avoid confusion, these facts are not added as simplification rules by default.

**lemma** *hurwitz-zeta-1*: *hurwitz-zeta c 1* = 0

**by** (*simp add: hurwitz-zeta-def*)

**lemma** *zeta-1*: *zeta 1* = 0

**by** (*simp add: zeta-def hurwitz-zeta-1*)

**lemma** *zeta-minus-pole-eq*:  $s \neq 1 \implies zeta\ s - 1 / (s - 1) = pre-zeta\ 1\ s$

**by** (*simp add: zeta-def hurwitz-zeta-def*)

**context**

**begin**

**private lemma** *holomorphic-pre-zeta-aux'*:

**assumes**  $a > 0$  *bounded U open U*  $U \subseteq \{s. Re\ s > \sigma\}$  **and**  $\sigma: \sigma > -2 * real\ n$

**shows** *pre-zeta-aux n a holomorphic-on U* **unfolding** *pre-zeta-aux-def*

**proof** (*intro holomorphic-intros*)

**define**  $C :: real$  **where**  $C = max\ 0\ (Sup\ ((\lambda s. norm\ (pochhammer\ s\ (Suc\ (2 * n))))\ 'closure\ U))$

**have** *compact (closure U)*

**using** *assms* **by** (*auto simp: compact-eq-bounded-closed*)

**hence** *compact ((\lambda s. norm (pochhammer s (Suc (2 \* n)))) 'closure U)*

**by** (*rule compact-continuous-image [rotated]*) (*auto intro!: continuous-intros*)

**hence** *bounded ((\lambda s. norm (pochhammer s (Suc (2 \* n)))) 'closure U)*

**by** (*simp add: compact-eq-bounded-closed*)

**hence**  $C: cmod\ (pochhammer\ s\ (Suc\ (2 * n))) \leq C$  **if**  $s \in U$  **for**  $s$

**using** *that closure-subset[of U]* **unfolding** *C-def*

**by** (*intro max.coboundedI2 cSup-upper bounded-imp-bdd-above*) (*auto simp: image-iff*)

**have**  $C' [simp]: C \geq 0$  **by** (*simp add: C-def*)

**let**  $?g = \lambda(x::real). C * (x + a) powr\ (-1 - 2 * of-nat\ n - \sigma)$

**let**  $?G = \lambda(x::real). C / (-2 * of-nat\ n - \sigma) * (x + a) powr\ (-2 * of-nat\ n - \sigma)$

**define** *poch'* **where**  $poch' = deriv\ (\lambda z::complex. pochhammer\ z\ (Suc\ (2 * n)))$

**have** [*derivative-intros*]:

$((\lambda z. pochhammer\ z\ (Suc\ (2 * n)))\ has-field-derivative\ poch'\ z)$  (*at z within A*)

**for**  $z :: complex$  **and**  $A$  **unfolding** *poch'-def*

**by** (*rule holomorphic-derivI [OF holomorphic-pochhammer [of - UNIV]]*) *auto*



**have**  $A$ : *continuous-on*  $A$  *poch'* **for**  $A$  **unfolding** *poch'-def*  
**by** (*rule continuous-on-subset*[*OF - subset-UNIV*],  
*intro holomorphic-on-imp-continuous-on holomorphic-deriv*)  
*(auto intro: holomorphic-pochhammer)*  
**note** [*continuous-intros*] = *continuous-on-compose2*[*OF this - subset-UNIV*]

**define**  $f'$  **where**  $f' = (\lambda z t. - (\text{poch}' z * \text{complex-of-real } (t + a) \text{ powr } (- 1 - 2 * \text{of-nat } n - z) -$   
 $Ln (\text{complex-of-real } (t + a)) * \text{complex-of-real } (t + a) \text{ powr } (- 1 - 2 * \text{of-nat } n - z) * \text{pochhammer } z (\text{Suc } (2 * n))))$

**show** ( $\lambda z. \text{EM-remainder } (\text{Suc } (2 * n)) (\lambda x. - (\text{pochhammer } z (\text{Suc } (2 * n)) * \text{complex-of-real } (x + a) \text{ powr } (- 1 - 2 * \text{of-nat } n - z))) 0$ )  
*holomorphic-on*  
**U unfolding** *pre-zeta-aux-def*  
**proof** (*rule holomorphic-EM-remainder*[*of - ?G ?g - - f'*], *goal-cases*)  
**case** (1  $x$ )  
**show** *?case*  
**by** (*insert 1*  $\sigma \langle a > 0 \rangle$ , *rule derivative-eq-intros refl | simp*) +  
*(auto simp: field-simps powr-diff powr-add powr-minus)*  
**next**  
**case** (2  $z t x$ )  
**note** [*derivative-intros*] = *has-field-derivative-powr-right* [*THEN DERIV-chain2*]  
**show** *?case*  
**by** (*insert 2*  $\sigma \langle a > 0 \rangle$ , (*rule derivative-eq-intros refl | (simp add: add-eq-0-iff; fail)*)) +  
*(simp add: f'-def)*  
**next**  
**case** 3  
**hence**  $*$ : *complex-of-real*  $x + \text{complex-of-real } a \notin \mathbb{R}_{\leq 0}$  **if**  $x \geq 0$  **for**  $x$   
**using** *nonpos-Reals-of-real-iff*[*of x+a, unfolded of-real-add*] **that**  $\langle a > 0 \rangle$  **by**  
*auto*  
**show** *?case using*  $\langle a > 0 \rangle$  **and**  $*$  **unfolding** *f'-def*  
**by** (*auto simp: case-prod-unfold add-eq-0-iff intro!: continuous-intros*)  
**next**  
**case** (4  $b c z e$ )  
**have**  $- 2 * \text{real } n < \sigma$  **by** (*fact*  $\sigma$ )  
**also from** 4 *assms* **have**  $\sigma < \text{Re } z$  **by** *auto*  
**finally show** *?case using* *assms 4*  
**by** (*intro integrable-continuous-real continuous-intros*) (*auto simp: add-eq-0-iff*)  
**next**  
**case** (5  $t x s$ )  
**thus** *?case using*  $\langle a > 0 \rangle$   
**by** (*intro integrable-EM-remainder'*) (*auto intro!: continuous-intros simp: add-eq-0-iff*)  
**next**  
**case** 6  
**from**  $\sigma$  **have** ( $\lambda y. C / (-2 * \text{real } n - \sigma) * (a + y) \text{ powr } (-2 * \text{real } n - \sigma))$   
 $\longrightarrow 0$

```

    by (intro tendsto-mult-right-zero tendsto-neg-pow
        filterlim-real-sequentially filterlim-tendsto-add-at-top [OF tendsto-const])
  auto
  thus ?case unfolding convergent-def by (auto simp: add-ac)
next
case 7
show ?case
proof (intro eventually-mono [OF eventually-ge-at-top[of 1]] ballI)
  fix x :: real and s :: complex assume x:  $x \geq 1$  and s:  $s \in U$ 
  have norm (- (pochhammer s (Suc (2 * n)) * of-real (x + a) powr (- 1 -
2 * of-nat n - s))) =
    norm (pochhammer s (Suc (2 * n)) * (x + a) powr (-1 - 2 * of-nat
n - Re s))
  (is ?N = -) using 7 <a > 0> x by (simp add: norm-mult norm-powr-real-powr)
  also have ...  $\leq$  ?g x
  using 7 assms x s <a > 0> by (intro mult-mono C powr-mono) auto
  finally show ?N  $\leq$  ?g x .
qed
qed (insert assms, auto)
qed (insert assms, auto)

```

**lemma** *analytic-pre-zeta-aux:*

```

  assumes a > 0
  shows pre-zeta-aux n a analytic-on {s. Re s > - 2 * real n}
  unfolding analytic-on-def
proof
  fix s assume s:  $s \in \{s. \text{Re } s > - 2 * \text{real } n\}$ 
  define  $\sigma$  where  $\sigma = (\text{Re } s - 2 * \text{real } n) / 2$ 
  with s have  $\sigma: \sigma > - 2 * \text{real } n$ 
  by (simp add:  $\sigma$ -def field-simps)
  from s have s':  $s \in \{s. \text{Re } s > \sigma\}$ 
  by (auto simp:  $\sigma$ -def field-simps)

  have open {s. Re s >  $\sigma$ }
  by (rule open-halfspace-Re-gt)
  with s' obtain  $\varepsilon$  where  $\varepsilon > 0$  ball s  $\varepsilon \subseteq \{s. \text{Re } s > \sigma\}$ 
  unfolding open-contains-ball by blast
  with  $\sigma$  have pre-zeta-aux n a holomorphic-on ball s  $\varepsilon$ 
  by (intro holomorphic-pre-zeta-aux' [OF assms, of -  $\sigma$ ]) auto
  with < $\varepsilon > 0$ > show  $\exists e > 0. \text{pre-zeta-aux } n \text{ a holomorphic-on ball } s \ e$ 
  by blast
qed
end

```

**context**

```

  fixes s :: complex and N :: nat and  $\zeta$  :: complex  $\Rightarrow$  complex and a :: real
  assumes s:  $\text{Re } s > 1$  and a:  $a > 0$ 
  defines  $\zeta \equiv (\lambda s. \sum n. (\text{of-nat } n + \text{of-real } a) \text{ powr } -s)$ 
begin

```

```

interpretation ζ: euler-maclaurin-nat'
  λx. of-real (x + a) powr (1 - s) / (1 - s) λx. of-real (x + a) powr -s
  λn x. (-1) ^ n * pochhammer s n * of-real (x + a) powr -(s + n)
  0 N ζ s {}
proof (standard, goal-cases)
  case 2
  show ?case by (simp add: powr-minus field-simps)
next
  case (3 k)
  have complex-of-real x + complex-of-real a = 0 ↔ x = -a for x
    by (simp only: of-real-add [symmetric] of-real-eq-0-iff add-eq-0-iff2)
  with a s show ?case
    by (intro continuous-intros) (auto simp: add-nonneg-nonneg)
next
  case (4 k x)
  with a have 0 < x + a by simp
  hence *: complex-of-real x + complex-of-real a ∉ ℝ≤0
    using nonpos-Reals-of-real-iff[of x+a, unfolded of-real-add] by auto
  have **: pochhammer z (Suc n) = - pochhammer z n * (-z - of-nat n ::
complex) for z n
    by (simp add: pochhammer-rec' field-simps)
  show ((λx. (-1) ^ k * pochhammer s k * of-real (x + a) powr - (s + of-nat
k))
    has-vector-derivative (-1) ^ Suc k * pochhammer s (Suc k) *
of-real (x + a) powr - (s + of-nat (Suc k))) (at x)
    by (insert 4 *, (rule has-vector-derivative-real-field derivative-eq-intros refl |
simp)+)
    (auto simp: divide-simps powr-add powr-diff powr-minus **)
next
  case 5
  with s a show ?case
    by (auto intro!: continuous-intros simp: minus-equation-iff add-eq-0-iff)
next
  case (6 x)
  with a have 0 < x + a by simp
  hence *: complex-of-real x + complex-of-real a ∉ ℝ≤0
    using nonpos-Reals-of-real-iff[of x+a, unfolded of-real-add] by auto
  show ?case unfolding of-real-add
    by (insert 6 s *, (rule has-vector-derivative-real-field derivative-eq-intros refl |
force simp add: minus-equation-iff)+)
next
  case 7
  from s a have (λk. (of-nat k + of-real a) powr -s) sums ζ s
    unfolding ζ-def by (intro summable-sums summable-hurwitz-zeta) auto
  hence 1: (λb. (∑ k=0..b. (of-nat k + of-real a) powr -s)) → ζ s
    by (simp add: sums-def')
  {

```

```

fix z assume Re z < 0
hence ((λb. (a + real b) powr Re z) → 0) at-top
by (intro tendsto-neg-powr filterlim-tendsto-add-at-top filterlim-real-sequentially)
auto
also have (λb. (a + real b) powr Re z) = (λb. norm ((of-nat b + a) powr z))
  using a by (subst norm-powr-real-powr) (auto simp: add-ac)
finally have ((λb. (of-nat b + a) powr z) → 0) at-top
  by (subst (asm) tendsto-norm-zero-iff) simp
} note * = this
have (λb. (of-nat b + a) powr (1 - s) / (1 - s)) → 0 / (1 - s)
  using s by (intro tendsto-divide tendsto-const *) auto
hence 2: (λb. (of-nat b + a) powr (1 - s) / (1 - s)) → 0
  by simp

have (λb. (∑ i<2 * N + 1. (bernoulli' (Suc i) / fact (Suc i)) *R
  ((- 1) ^ i * pochhammer s i * (of-nat b + a) powr -(s + of-nat i))))
  → (∑ i<2 * N + 1. (bernoulli' (Suc i) / fact (Suc i)) *R
  ((- 1) ^ i * pochhammer s i * 0))
  using s by (intro tendsto-intros *) auto
hence 3: (λb. (∑ i<2 * N + 1. (bernoulli' (Suc i) / fact (Suc i)) *R
  ((- 1) ^ i * pochhammer s i * (of-nat b + a) powr -(s + of-nat i))))
  → 0
  by simp

from tendsto-diff[OF tendsto-diff[OF 1 2] 3]
show ?case by simp
qed simp-all

The pre-ζ functions agree with the infinite sum that is used to define the ζ
function for ℞(s) > 1.

lemma pre-zeta-aux-conv-zeta:
  pre-zeta-aux N a s = ζ s + a powr (1 - s) / (1 - s)
proof -
  let ?R = (∑ i=1..N. ((bernoulli (2*i) / fact (2*i)) *R pochhammer s (2*i -
  1) * of-real a powr (-s - (2*i-1))))
  let ?S = EM-remainder (Suc (2 * N)) (λx. - (pochhammer s (Suc (2*N)) *
  of-real (x + a) powr (- 1 - 2 * of-nat N - s))) 0
  from ζ.euler-maclaurin-strong-nat'[OF le-reft, simplified]
  have of-real a powr -s = a powr (1 - s) / (1 - s) + ζ s + a powr -s / 2 +
  (-?R) - ?S
  unfolding sum-negf [symmetric] by (simp add: scaleR-conv-of-real pre-zeta-aux-def
  mult-ac)
  thus ?thesis unfolding pre-zeta-aux-def

  by (simp add: field-simps del: div-mult-self3 div-mult-self4 div-mult-self2 div-mult-self1)
qed

end

```

Since all of the partial pre- $\zeta$  functions are analytic and agree in the halfspace with  $\Re(s) > 0$ , they must agree in their entire domain.

**lemma** *pre-zeta-aux-eq*:

**assumes**  $m \leq n$   $a > 0$   $\Re s > -2 * \text{real } m$

**shows**  $\text{pre-zeta-aux } m \ a \ s = \text{pre-zeta-aux } n \ a \ s$

**proof** –

**have**  $\text{pre-zeta-aux } n \ a \ s - \text{pre-zeta-aux } m \ a \ s = 0$

**proof** (*rule analytic-continuation[of  $\lambda s. \text{pre-zeta-aux } n \ a \ s - \text{pre-zeta-aux } m \ a \ s]$* )

**show** ( $\lambda s. \text{pre-zeta-aux } n \ a \ s - \text{pre-zeta-aux } m \ a \ s$ ) *holomorphic-on*  $\{s. \Re s > -2 * \text{real } m\}$

**using** *assms* **by** (*intro holomorphic-intros analytic-imp-holomorphic analytic-on-subset[OF analytic-pre-zeta-aux]*) *auto*

**next**

**fix**  $s$  **assume**  $s \in \{s. \Re s > 1\}$

**with**  $\langle a > 0 \rangle$  **show**  $\text{pre-zeta-aux } n \ a \ s - \text{pre-zeta-aux } m \ a \ s = 0$

**by** (*simp add: pre-zeta-aux-conv-zeta*)

**next**

**have**  $2 \in \{s. \Re s > 1\}$  **by** *simp*

**also have**  $\dots = \text{interior } \dots$

**by** (*intro interior-open [symmetric] open-halfspace-Re-gt*)

**finally show**  $2 \text{ islimpt } \{s. \Re s > 1\}$

**by** (*rule interior-limit-point*)

**next**

**show** *connected*  $\{s. \Re s > -2 * \text{real } m\}$

**using** *convex-halfspace-gt[of  $-2 * \text{real } m \ 1::\text{complex}$ ]*

**by** (*intro convex-connected*) *auto*

**qed** (*insert assms, auto simp: open-halfspace-Re-gt*)

**thus** *?thesis* **by** *simp*

**qed**

**lemma** *pre-zeta-aux-eq'*:

**assumes**  $a > 0$   $\Re s > -2 * \text{real } m$   $\Re s > -2 * \text{real } n$

**shows**  $\text{pre-zeta-aux } m \ a \ s = \text{pre-zeta-aux } n \ a \ s$

**proof** (*cases m n rule: linorder-cases*)

**case** *less*

**with** *assms* **show** *?thesis* **by** (*intro pre-zeta-aux-eq*) *auto*

**next**

**case** *greater*

**with** *assms* **show** *?thesis* **by** (*subst eq-commute, intro pre-zeta-aux-eq*) *auto*

**qed** *auto*

**lemma** *pre-zeta-aux-eq-pre-zeta*:

**assumes**  $\Re s > -2 * \text{real } n$  **and**  $a > 0$

**shows**  $\text{pre-zeta-aux } n \ a \ s = \text{pre-zeta } a \ s$

**unfolding** *pre-zeta-def*

**proof** (*intro pre-zeta-aux-eq'*)

**from** *assms* **show**  $-2 * \text{real } ( \text{nat } (1 - \lceil \Re s / 2 \rceil) ) < \Re s$

**by** *linarith*

**qed** (*insert assms, simp-all*)

This means that the idea of iterating that construction infinitely does yield a well-defined entire function.

**lemma** *analytic-pre-zeta*:

**assumes**  $a > 0$

**shows** *pre-zeta a analytic-on A*

**unfolding** *analytic-on-def*

**proof**

**fix**  $s$  **assume**  $s \in A$

**let**  $?B = \{s'. \text{Re } s' > \text{of-int } \lfloor \text{Re } s \rfloor - 1\}$

**have**  $s: s \in ?B$  **by** *simp linarith?*

**moreover have** *open ?B by (rule open-halfspace-Re-gt)*

**ultimately obtain**  $\varepsilon$  **where**  $\varepsilon: \varepsilon > 0$  *ball s  $\varepsilon \subseteq ?B$*

**unfolding** *open-contains-ball by blast*

**define**  $C$  **where**  $C = \text{ball } s \ \varepsilon$

**note** *analytic = analytic-on-subset[OF analytic-pre-zeta-aux]*

**have** *pre-zeta-aux (nat  $\lfloor - \text{Re } s \rfloor + 2$ ) a holomorphic-on C*

**proof** (*intro analytic-imp-holomorphic analytic subsetI assms, goal-cases*)

**case** (1  $w$ )

**with**  $\varepsilon$  **have**  $w \in ?B$  **by** (*auto simp: C-def*)

**thus**  $?case$  **by** (*auto simp: ceiling-minus*)

**qed**

**also have**  $?this \iff \text{pre-zeta } a \text{ holomorphic-on } C$

**proof** (*intro holomorphic-cong refl pre-zeta-aux-eq-pre-zeta assms*)

**fix**  $w$  **assume**  $w \in C$

**with**  $\varepsilon$  **have**  $w: w \in ?B$  **by** (*auto simp: C-def*)

**thus**  $-2 * \text{real } (\text{nat } \lfloor - \text{Re } s \rfloor + 2) < \text{Re } w$

**by** (*simp add: ceiling-minus*)

**qed**

**finally show**  $\exists e > 0. \text{pre-zeta } a \text{ holomorphic-on ball } s \ e$

**using**  $\langle \varepsilon > 0 \rangle$  **unfolding** *C-def by blast*

**qed**

**lemma** *holomorphic-pre-zeta [holomorphic-intros]*:

*f holomorphic-on A  $\implies a > 0 \implies (\lambda z. \text{pre-zeta } a (f z)) \text{ holomorphic-on } A$*

**using** *holomorphic-on-compose [OF - analytic-imp-holomorphic [OF analytic-pre-zeta], of f]*

**by** (*simp add: o-def*)

**corollary** *continuous-on-pre-zeta*:

$a > 0 \implies \text{continuous-on } A (\text{pre-zeta } a)$

**by** (*intro holomorphic-on-imp-continuous-on holomorphic-intros auto*)

**corollary** *continuous-on-pre-zeta' [continuous-intros]*:

*continuous-on A f  $\implies a > 0 \implies \text{continuous-on } A (\lambda x. \text{pre-zeta } a (f x))$*

**using** *continuous-on-compose2 [OF continuous-on-pre-zeta, of a A f f ' A]*

**by** (*auto simp: image-iff*)

**corollary** *continuous-pre-zeta* [*continuous-intros*]:  
 $a > 0 \implies \text{continuous (at } s \text{ within } A) \text{ (pre-zeta } a)$   
**by** (*rule continuous-within-subset*[*of - UNIV*])  
(*insert continuous-on-pre-zeta*[*of a UNIV*],  
*auto simp: continuous-on-eq-continuous-at open-Compl*)

**corollary** *continuous-pre-zeta'* [*continuous-intros*]:  
 $a > 0 \implies \text{continuous (at } s \text{ within } A) f \implies$   
 $\text{continuous (at } s \text{ within } A) (\lambda s. \text{pre-zeta } a (f s))$   
**using** *continuous-within-compose3*[*OF continuous-pre-zeta, of a s A f*] **by** *auto*

It is now obvious that  $\zeta$  is holomorphic everywhere except 1, where it has a simple pole with residue 1, which we can simply read off.

**theorem** *holomorphic-hurwitz-zeta*:  
**assumes**  $a > 0 \ 1 \notin A$   
**shows** *hurwitz-zeta a holomorphic-on A*  
**proof** –  
**have** ( $\lambda s. \text{pre-zeta } a \ s + \text{complex-of-real } a \ \text{powr } (1 - s) / (s - 1)$ ) *holomorphic-on A*  
**using** *assms by (auto intro!: holomorphic-intros)*  
**also from** *assms have ?this  $\longleftrightarrow$  ?thesis*  
**by** (*intro holomorphic-cong*) (*auto simp: hurwitz-zeta-def*)  
**finally show** *?thesis .*  
**qed**

**corollary** *holomorphic-hurwitz-zeta'* [*holomorphic-intros*]:  
**assumes**  $f \text{ holomorphic-on } A$  **and**  $a > 0$  **and**  $\bigwedge z. z \in A \implies f z \neq 1$   
**shows**  $(\lambda x. \text{hurwitz-zeta } a (f x)) \text{ holomorphic-on } A$   
**proof** –  
**have** *hurwitz-zeta a  $\circ$  f holomorphic-on A using assms*  
**by** (*intro holomorphic-on-compose-gen*[*of - - - f ' A*] *holomorphic-hurwitz-zeta assms*) *auto*  
**thus** *?thesis by (simp add: o-def)*  
**qed**

**theorem** *holomorphic-zeta*:  $1 \notin A \implies \text{zeta holomorphic-on } A$   
**unfolding** *zeta-def by (auto intro: holomorphic-intros)*

**corollary** *holomorphic-zeta'* [*holomorphic-intros*]:  
**assumes**  $f \text{ holomorphic-on } A$  **and**  $\bigwedge z. z \in A \implies f z \neq 1$   
**shows**  $(\lambda x. \text{zeta } (f x)) \text{ holomorphic-on } A$   
**using** *assms unfolding zeta-def by (auto intro: holomorphic-intros)*

**corollary** *analytic-hurwitz-zeta*:  
**assumes**  $a > 0 \ 1 \notin A$   
**shows** *hurwitz-zeta a analytic-on A*  
**proof** –  
**from** *assms(1) have hurwitz-zeta a holomorphic-on  $-\{1\}$*

by (rule holomorphic-hurwitz-zeta) auto  
 also have ?this  $\longleftrightarrow$  hurwitz-zeta a analytic-on  $-\{1\}$   
 by (intro analytic-on-open [symmetric]) auto  
 finally show ?thesis by (rule analytic-on-subset) (insert assms, auto)  
 qed

**corollary analytic-zeta:**  $1 \notin A \implies$  zeta analytic-on  $A$   
 unfolding zeta-def by (rule analytic-hurwitz-zeta) auto

**corollary continuous-on-hurwitz-zeta:**  
 $a > 0 \implies 1 \notin A \implies$  continuous-on  $A$  (hurwitz-zeta a)  
 by (intro holomorphic-on-imp-continuous-on holomorphic-intros) auto

**corollary continuous-on-hurwitz-zeta'** [continuous-intros]:  
 $continuous\ on\ A\ f \implies a > 0 \implies (\bigwedge x. x \in A \implies f\ x \neq 1) \implies$   
 $continuous\ on\ A\ (\lambda x. hurwitz\ zeta\ a\ (f\ x))$   
 using continuous-on-compose2 [OF continuous-on-hurwitz-zeta, of a f ' A A f]  
 by (auto simp: image-iff)

**corollary continuous-on-zeta:**  $1 \notin A \implies$  continuous-on  $A$  zeta  
 by (intro holomorphic-on-imp-continuous-on holomorphic-intros) auto

**corollary continuous-on-zeta'** [continuous-intros]:  
 $continuous\ on\ A\ f \implies (\bigwedge x. x \in A \implies f\ x \neq 1) \implies$   
 $continuous\ on\ A\ (\lambda x. zeta\ (f\ x))$   
 using continuous-on-compose2 [OF continuous-on-zeta, of f ' A A f]  
 by (auto simp: image-iff)

**corollary continuous-hurwitz-zeta** [continuous-intros]:  
 $a > 0 \implies s \neq 1 \implies$  continuous (at  $s$  within  $A$ ) (hurwitz-zeta a)  
 by (rule continuous-within-subset[of - UNIV])  
 (insert continuous-on-hurwitz-zeta[of a  $-\{1\}$ ],  
 auto simp: continuous-on-eq-continuous-at open-Compl)

**corollary continuous-hurwitz-zeta'** [continuous-intros]:  
 $a > 0 \implies f\ s \neq 1 \implies$  continuous (at  $s$  within  $A$ )  $f \implies$   
 $continuous\ (at\ s\ within\ A)\ (\lambda s. hurwitz\ zeta\ a\ (f\ s))$   
 using continuous-within-compose3[OF continuous-hurwitz-zeta, of a f s A] by  
 auto

**corollary continuous-zeta** [continuous-intros]:  
 $s \neq 1 \implies$  continuous (at  $s$  within  $A$ ) zeta  
 unfolding zeta-def by (intro continuous-intros) auto

**corollary continuous-zeta'** [continuous-intros]:  
 $f\ s \neq 1 \implies$  continuous (at  $s$  within  $A$ )  $f \implies$  continuous (at  $s$  within  $A$ )  $(\lambda s. zeta$   
 $(f\ s))$   
 unfolding zeta-def by (intro continuous-intros) auto



**corollary** *field-differentiable-at-zeta:*

**assumes**  $s \neq 1$

**shows** *zeta field-differentiable at s*

**proof** –

**have** *zeta holomorphic-on* ( $-\{1\}$ ) **using** *holomorphic-zeta* **by force**

**moreover have** *open* ( $-\{1\} :: \text{complex set}$ ) **by** (*intro open-Compl*) *auto*

**ultimately show** *?thesis* **using** *assms*

**by** (*auto simp add: holomorphic-on-open open-halfspace-Re-gt open-Diff field-differentiable-def*)

**qed**

**theorem** *is-pole-hurwitz-zeta:*

**assumes**  $a > 0$

**shows** *is-pole* (*hurwitz-zeta a*) 1

**proof** –

**from** *assms* **have** *continuous-on UNIV* (*pre-zeta a*)

**by** (*intro holomorphic-on-imp-continuous-on analytic-imp-holomorphic analytic-pre-zeta*)

**hence** *isCont* (*pre-zeta a*) 1

**by** (*auto simp: continuous-on-eq-continuous-at*)

**hence** \*: *pre-zeta a*  $-1 \rightarrow$  *pre-zeta a* 1

**by** (*simp add: isCont-def*)

**from** *assms* **have** *isCont* ( $\lambda s. \text{complex-of-real } a \text{ powr } (1 - s)$ ) 1

**by** (*intro isCont-powr-complex*) *auto*

**with** *assms* **have** \*\*: ( $\lambda s. \text{complex-of-real } a \text{ powr } (1 - s)$ )  $-1 \rightarrow$  1

**by** (*simp add: isCont-def*)

**have** ( $\lambda s :: \text{complex}. s - 1$ )  $-1 \rightarrow 1 - 1$  **by** (*intro tendsto-intros*)

**hence** *filterlim* ( $\lambda s :: \text{complex}. s - 1$ ) (at 0) (at 1)

**by** (*auto simp: filterlim-at eventually-at-filter*)

**hence** \*\*\*: *filterlim* ( $\lambda s :: \text{complex}. a \text{ powr } (1 - s) / (s - 1)$ ) *at-infinity* (at 1)

**by** (*intro filterlim-divide-at-infinity [OF \*\*]*) *auto*

**have** *is-pole* ( $\lambda s. \text{pre-zeta } a \text{ } s + \text{complex-of-real } a \text{ powr } (1 - s) / (s - 1)$ ) 1

**unfolding** *is-pole-def hurwitz-zeta-def* **by** (*rule tendsto-add-filterlim-at-infinity*

\* \*\*\*)+

**also have** *?this*  $\longleftrightarrow$  *?thesis* **unfolding** *is-pole-def*

**by** (*intro filterlim-cong refl*) (*auto simp: eventually-at-filter hurwitz-zeta-def*)

**finally show** *?thesis* .

**qed**

**corollary** *is-pole-zeta: is-pole zeta 1*

**by** (*simp add: is-pole-hurwitz-zeta zeta-def*)

**theorem** *zorder-hurwitz-zeta:*

**assumes**  $a > 0$

**shows** *zorder* (*hurwitz-zeta a*) 1 = -1

**proof** (*rule zorder-eqI*[of UNIV])

**fix**  $w :: \text{complex}$  **assume**  $w \neq 1$

**thus** *hurwitz-zeta a w* = (*pre-zeta a w* \* ( $w - 1$ ) + *powr* (1 - w)) \* ( $w - 1$ )  
*powi -1*

**by** (*auto simp add: hurwitz-zeta-def field-simps*)

**qed** (use *assms* in  $\langle$ auto intro!: holomorphic-intros $\rangle$ )

**corollary** *zorder-zeta*:  $\text{zorder zeta } 1 = - 1$   
**unfolding** *zeta-def* **by** (rule *zorder-hurwitz-zeta*) *auto*

**theorem** *residue-hurwitz-zeta*:

**assumes**  $a > 0$

**shows**  $\text{residue (hurwitz-zeta } a) 1 = 1$

**proof** –

**note** *holo* = *analytic-imp-holomorphic*[*OF analytic-pre-zeta*]

**have**  $\text{residue (hurwitz-zeta } a) 1 = \text{residue } (\lambda z. \text{pre-zeta } a z + a \text{ powr } (1 - z) / (z - 1)) 1$

**by** (*intro residue-cong*) (*auto simp: eventually-at-filter hurwitz-zeta-def*)

**also have**  $\dots = \text{residue } (\lambda z. a \text{ powr } (1 - z) / (z - 1)) 1$  **using** *assms*

**by** (*subst residue-add* [*of UNIV*])

(*auto intro!: holomorphic-intros holo intro: residue-holo*[*of UNIV, OF - - holo*])

**also have**  $\dots = \text{complex-of-real } a \text{ powr } (1 - 1)$

**using** *assms* **by** (*intro residue-simple* [*of UNIV*]) (*auto intro!: holomorphic-intros*)

**also from** *assms* **have**  $\dots = 1$  **by** *simp*

**finally show** *?thesis* .

**qed**

**corollary** *residue-zeta*:  $\text{residue zeta } 1 = 1$   
**unfolding** *zeta-def* **by** (rule *residue-hurwitz-zeta*) *auto*

**lemma** *zeta-bigo-at-1*:  $\text{zeta} \in O[\text{at } 1 \text{ within } A](\lambda x. 1 / (x - 1))$

**proof** –

**have**  $\text{zeta} \in \Theta[\text{at } 1 \text{ within } A](\lambda s. \text{pre-zeta } 1 s + 1 / (s - 1))$

**by** (*intro bigthetaI-cong*) (*auto simp: eventually-at-filter zeta-def hurwitz-zeta-def*)

**also have**  $(\lambda s. \text{pre-zeta } 1 s + 1 / (s - 1)) \in O[\text{at } 1 \text{ within } A](\lambda s. 1 / (s - 1))$

**proof** (rule *sum-in-bigo*)

**have** *continuous-on UNIV* (*pre-zeta 1*)

**by** (*intro holomorphic-on-imp-continuous-on holomorphic-intros*) *auto*

**hence** *isCont* (*pre-zeta 1*) 1 **by** (*auto simp: continuous-on-eq-continuous-at*)

**hence** *continuous* (*at 1 within A*) (*pre-zeta 1*)

**by** (rule *continuous-within-subset*) *auto*

**hence** *pre-zeta 1*  $\in O[\text{at } 1 \text{ within } A](\lambda-. 1)$

**by** (*intro continuous-imp-bigo-1*) *auto*

**also have** *ev: eventually* ( $\lambda s. s \in \text{ball } 1 1 \wedge s \neq 1 \wedge s \in A$ ) (*at 1 within A*)

**by** (*intro eventually-at-ball'*) *auto*

**have**  $(\lambda-. 1) \in O[\text{at } 1 \text{ within } A](\lambda s. 1 / (s - 1))$

**by** (*intro landau-o.bigI*[*of 1*] *eventually-mono*[*OF ev*])

(*auto simp: eventually-at-filter norm-divide dist-norm norm-minus-commute field-simps*)

**finally show** *pre-zeta 1*  $\in O[\text{at } 1 \text{ within } A](\lambda s. 1 / (s - 1))$  .

**qed** *simp-all*

**finally show** *?thesis* .

**qed**

**theorem**

**assumes**  $a > 0$   $Re\ s > 1$

**shows** *hurwitz-zeta-conv-suminf*:  $hurwitz-zeta\ a\ s = (\sum n. (of-nat\ n + of-real\ a)^{powr\ -s})$

**and** *sums-hurwitz-zeta*:  $(\lambda n. (of-nat\ n + of-real\ a)^{powr\ -s})\ sums\ hurwitz-zeta\ a\ s$

**proof** –

**from** *assms* **have** [*simp*]:  $s \neq 1$  **by** *auto*

**from** *assms* **have**  $hurwitz-zeta\ a\ s = pre-zeta-aux\ 0\ a\ s + of-real\ a\ powr\ (1 - s) / (s - 1)$

**by** (*simp add: hurwitz-zeta-def pre-zeta-def*)

**also from** *assms* **have**  $pre-zeta-aux\ 0\ a\ s = (\sum n. (of-nat\ n + of-real\ a)^{powr\ -s}) +$

$$of-real\ a\ powr\ (1 - s) / (1 - s)$$

**by** (*intro pre-zeta-aux-conv-zeta*)

**also have**  $\dots + a\ powr\ (1 - s) / (s - 1) =$

$$(\sum n. (of-nat\ n + of-real\ a)^{powr\ -s}) + a\ powr\ (1 - s) * (1 / (1 - s) + 1 / (s - 1))$$

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

**also have**  $1 / (1 - s) + 1 / (s - 1) = 0$

**by** (*simp add: divide-simps*)

**finally show**  $hurwitz-zeta\ a\ s = (\sum n. (of-nat\ n + of-real\ a)^{powr\ -s})$  **by** *simp*

**moreover have**  $(\lambda n. (of-nat\ n + of-real\ a)^{powr\ -s})\ sums\ (\sum n. (of-nat\ n + of-real\ a)^{powr\ -s})$

**by** (*intro summable-sums summable-hurwitz-zeta assms*)

**ultimately show**  $(\lambda n. (of-nat\ n + of-real\ a)^{powr\ -s})\ sums\ hurwitz-zeta\ a\ s$

**by** *simp*

**qed**

**corollary**

**assumes**  $Re\ s > 1$

**shows** *zeta-conv-suminf*:  $zeta\ s = (\sum n. of-nat\ (Suc\ n)^{powr\ -s})$

**and** *sums-zeta*:  $(\lambda n. of-nat\ (Suc\ n)^{powr\ -s})\ sums\ zeta\ s$

**using** *hurwitz-zeta-conv-suminf*[*of 1 s*] *sums-hurwitz-zeta*[*of 1 s*] *assms*

**by** (*simp-all add: zeta-def add-ac*)

**corollary**

**assumes**  $n > 1$

**shows** *zeta-nat-conv-suminf*:  $zeta\ (of-nat\ n) = (\sum k. 1 / of-nat\ (Suc\ k)^{\wedge n})$

**and** *sums-zeta-nat*:  $(\lambda k. 1 / of-nat\ (Suc\ k)^{\wedge n})\ sums\ zeta\ (of-nat\ n)$

**proof** –

**have**  $(\lambda k. of-nat\ (Suc\ k)^{powr\ -of-nat\ n})\ sums\ zeta\ (of-nat\ n)$

**using** *assms* **by** (*intro sums-zeta auto*)

**also have**  $(\lambda k. of-nat\ (Suc\ k)^{powr\ -of-nat\ n}) = (\lambda k. 1 / of-nat\ (Suc\ k)^{\wedge n} :: complex)$

**by** (*simp add: powr-minus divide-simps del: of-nat-Suc*)

**finally show**  $(\lambda k. 1 / of-nat\ (Suc\ k)^{\wedge n})\ sums\ zeta\ (of-nat\ n)$ .

**thus**  $zeta\ (of-nat\ n) = (\sum k. 1 / of-nat\ (Suc\ k)^{\wedge n})$  **by** (*simp add: sums-iff*)

qed

**lemma** *pre-zeta-aux-cnj* [simp]:

**assumes**  $a > 0$

**shows**  $\text{pre-zeta-aux } n \ a \ (\text{cnj } z) = \text{cnj } (\text{pre-zeta-aux } n \ a \ z)$

**proof** –

**have**  $\text{cnj } (\text{pre-zeta-aux } n \ a \ z) =$

$\text{of-real } a \ \text{powr } -\text{cnj } z / 2 + (\sum_{x=1..n}. (\text{bernoulli } (2 * x) / \text{fact } (2 * x)))$

$*_R$

$a \ \text{powr } (-\text{cnj } z - (2*x-1)) * \text{pochhammer } (\text{cnj } z) (2*x-1) +$

$\text{EM-remainder } (2*n+1)$

$(\lambda x. -(\text{pochhammer } (\text{cnj } z) (\text{Suc } (2 * n)) *$

$\text{cnj } (\text{of-real } (x + a) \ \text{powr } (-1 - 2 * \text{of-nat } n - z)))) 0$

**(is - = - + ?A + ?B) unfolding** *pre-zeta-aux-def complex-cnj-add* **using** *assms*

**by** (*subst EM-remainder-cnj [symmetric]*)

(*auto intro!: continuous-intros simp: cnj-powr add-eq-0-iff mult-ac*)

**also have**  $?B = \text{EM-remainder } (2*n+1)$

$(\lambda x. -(\text{pochhammer } (\text{cnj } z) (\text{Suc } (2 * n)) * \text{of-real } (x + a) \ \text{powr } (-1 - 2$

$* \ \text{of-nat } n - \text{cnj } z))) 0$

**using** *assms* **by** (*intro EM-remainder-cong*) (*auto simp: cnj-powr*)

**also have**  $\text{of-real } a \ \text{powr } -\text{cnj } z / 2 + ?A + \dots = \text{pre-zeta-aux } n \ a \ (\text{cnj } z)$

**by** (*simp add: pre-zeta-aux-def mult-ac*)

**finally show** *?thesis ..*

qed

**lemma** *pre-zeta-cnj* [simp]:  $a > 0 \implies \text{pre-zeta } a \ (\text{cnj } z) = \text{cnj } (\text{pre-zeta } a \ z)$

**by** (*simp add: pre-zeta-def*)

**lemma** *hurwitz-zeta-cnj* [simp]:  $a > 0 \implies \text{hurwitz-zeta } a \ (\text{cnj } z) = \text{cnj } (\text{hurwitz-zeta } a \ z)$

**proof** –

**assume**  $a > 0$

**moreover have**  $\text{cnj } z = 1 \iff z = 1$  **by** (*simp add: complex-eq-iff*)

**ultimately show** *?thesis* **by** (*auto simp: hurwitz-zeta-def cnj-powr*)

qed

**lemma** *zeta-cnj* [simp]:  $\text{zeta } (\text{cnj } z) = \text{cnj } (\text{zeta } z)$

**by** (*simp add: zeta-def*)

**corollary** *hurwitz-zeta-real*:  $a > 0 \implies \text{hurwitz-zeta } a \ (\text{of-real } x) \in \mathbb{R}$

**using** *hurwitz-zeta-cnj [of a of-real x]* **by** (*simp add: Reals-cnj-iff del: zeta-cnj*)

**corollary** *zeta-real*:  $\text{zeta } (\text{of-real } x) \in \mathbb{R}$

**unfolding** *zeta-def* **by** (*rule hurwitz-zeta-real*) *auto*

**corollary** *zeta-real'*:  $z \in \mathbb{R} \implies \text{zeta } z \in \mathbb{R}$

**by** (*elim Reals-cases*) (*auto simp: zeta-real*)

## 2.3 Connection to Dirichlet series

**lemma** *eval-fds-zeta*:  $Re\ s > 1 \implies eval-fds\ fds-zeta\ s = zeta\ s$   
**using** *sums-zeta* [of *s*] **by** (*intro eval-fds-eqI*) (*auto simp: powr-minus divide-simps*)

**theorem** *euler-product-zeta*:

**assumes**  $Re\ s > 1$

**shows**  $(\lambda n. \prod_{p \leq n}. \text{if prime } p \text{ then inverse } (1 - 1 / \text{of-nat } p \text{ powr } s) \text{ else } 1) \longrightarrow zeta\ s$

**using** *euler-product-fds-zeta*[of *s*] *assms* **unfolding** *nat-power-complex-def*  
**by** (*simp add: eval-fds-zeta*)

**corollary** *euler-product-zeta'*:

**assumes**  $Re\ s > 1$

**shows**  $(\lambda n. \prod_{p \mid \text{prime } p \wedge p \leq n}. \text{inverse } (1 - 1 / \text{of-nat } p \text{ powr } s)) \longrightarrow zeta\ s$

**proof** –

**note** *euler-product-zeta* [*OF assms*]

**also have**  $(\lambda n. \prod_{p \leq n}. \text{if prime } p \text{ then inverse } (1 - 1 / \text{of-nat } p \text{ powr } s) \text{ else } 1) =$

$(\lambda n. \prod_{p \mid \text{prime } p \wedge p \leq n}. \text{inverse } (1 - 1 / \text{of-nat } p \text{ powr } s))$

**by** (*intro ext prod.mono-neutral-cong-right refl*) *auto*

**finally show** *?thesis* .

**qed**

**theorem** *zeta-Re-gt-1-nonzero*:  $Re\ s > 1 \implies zeta\ s \neq 0$

**using** *eval-fds-zeta-nonzero*[of *s*] **by** (*simp add: eval-fds-zeta*)

**theorem** *tendsto-zeta-Re-going-to-at-top*:  $(zeta \longrightarrow 1)$  (*Re going-to at-top*)

**proof** (*rule Lim-transform-eventually*)

**have** *eventually*  $(\lambda x :: \text{real}. x > 1)$  *at-top*

**by** (*rule eventually-gt-at-top*)

**hence** *eventually*  $(\lambda s. Re\ s > 1)$  (*Re going-to at-top*)

**by** *blast*

**thus** *eventually*  $(\lambda z. eval-fds\ fds-zeta\ z = zeta\ z)$  (*Re going-to at-top*)

**by** *eventually-elim* (*simp add: eval-fds-zeta*)

**next**

**have** *conv-abscissa*  $(fds-zeta :: \text{complex } fds) \leq 1$

**proof** (*rule conv-abscissa-leI*)

**fix** *c'* **assume** *ereal*  $c' > 1$

**thus**  $\exists s. s \cdot 1 = c' \wedge fds\text{-converges } fds\text{-zeta } (s :: \text{complex})$

**by** (*auto intro!: exI*[of - of-real *c'*])

**qed**

**hence**  $(eval-fds\ fds-zeta \longrightarrow fds\text{-nth } fds-zeta\ 1)$  (*Re going-to at-top*)

**by** (*intro tendsto-eval-fds-Re-going-to-at-top'*) *auto*

**thus**  $(eval-fds\ fds-zeta \longrightarrow 1)$  (*Re going-to at-top*) **by** *simp*

**qed**

**lemma** *conv-abscissa-zeta* [*simp*]: *conv-abscissa*  $(fds-zeta :: \text{complex } fds) = 1$

**and** *abs-conv-abscissa-zeta* [*simp*]: *abs-conv-abscissa*  $(fds-zeta :: \text{complex } fds) =$

1

**proof** –

**let**  $?z = \text{fds-zeta} :: \text{complex fds}$

**have**  $A: \text{conv-abscissa } ?z \geq 1$

**proof** (*intro conv-abscissa-geI*)

**fix**  $c'$  **assume**  $\text{ereal } c' < 1$

**hence**  $\neg \text{summable } (\lambda n. \text{real } n \text{ powr } -c')$

**by** (*subst summable-real-powr-iff*) *auto*

**hence**  $\neg \text{summable } (\lambda n. \text{of-real } (\text{real } n \text{ powr } -c') :: \text{complex})$

**by** (*subst summable-of-real-iff*)

**also have**  $\text{summable } (\lambda n. \text{of-real } (\text{real } n \text{ powr } -c') :: \text{complex}) \longleftrightarrow$   
 $\text{fds-converges fds-zeta } (\text{of-real } c' :: \text{complex})$

**unfolding** *fds-converges-def*

**by** (*intro summable-cong eventually-mono [OF eventually-gt-at-top[of 0]]*)  
(*simp add: fds-nth-zeta powr-Reals-eq powr-minus divide-simps*)

**finally show**  $\exists s :: \text{complex}. s \cdot 1 = c' \wedge \neg \text{fds-converges fds-zeta } s$

**by** (*intro exI[of - of-real c']*) *auto*

**qed**

**have**  $B: \text{abs-conv-abscissa } ?z \leq 1$

**proof** (*intro abs-conv-abscissa-leI*)

**fix**  $c'$  **assume**  $1 < \text{ereal } c'$

**thus**  $\exists s :: \text{complex}. s \cdot 1 = c' \wedge \text{fds-abs-converges fds-zeta } s$

**by** (*intro exI[of - of-real c']*) *auto*

**qed**

**have**  $\text{conv-abscissa } ?z \leq \text{abs-conv-abscissa } ?z$

**by** (*rule conv-le-abs-conv-abscissa*)

**also note**  $B$

**finally show**  $\text{conv-abscissa } ?z = 1$  **using**  $A$  **by** (*intro antisym*)

**note**  $A$

**also have**  $\text{conv-abscissa } ?z \leq \text{abs-conv-abscissa } ?z$

**by** (*rule conv-le-abs-conv-abscissa*)

**finally show**  $\text{abs-conv-abscissa } ?z = 1$  **using**  $B$  **by** (*intro antisym*)

**qed**

**theorem** *deriv-zeta-sums*:

**assumes**  $s: \text{Re } s > 1$

**shows**  $(\lambda n. -\text{of-real } (\ln (\text{real } (\text{Suc } n)))) / \text{of-nat } (\text{Suc } n) \text{ powr } s \text{ sums deriv zeta}$   
 $s$

**proof** –

**from**  $s$  **have**  $\text{fds-converges } (\text{fds-deriv fds-zeta}) s$

**by** (*intro fds-converges-deriv simp-all*)

**with**  $s$  **have**  $(\lambda n. -\text{of-real } (\ln (\text{real } (\text{Suc } n)))) / \text{of-nat } (\text{Suc } n) \text{ powr } s \text{ sums}$   
 $\text{deriv } (\text{eval-fds fds-zeta}) s$

**unfolding** *fds-converges-altdef*

**by** (*simp add: fds-nth-deriv scaleR-conv-of-real eval-fds-deriv eval-fds-zeta*)

**also from**  $s$  **have** *eventually*  $(\lambda s. s \in \{s. \text{Re } s > 1\})$  (*nhds*  $s$ )

by (intro eventually-nhds-in-open) (auto simp: open-halfspace-Re-gt)  
 hence eventually  $(\lambda s. \text{eval-fds fds-zeta } s = \text{zeta } s)$  (nhds  $s$ )  
 by eventually-elim (auto simp: eval-fds-zeta)  
 hence deriv (eval-fds fds-zeta)  $s = \text{deriv zeta } s$   
 by (intro deriv-cong-ev refl)  
 finally show ?thesis .  
 qed

**theorem** inverse-zeta-sums:

assumes  $s: \text{Re } s > 1$   
 shows  $(\lambda n. \text{moebius-mu } (\text{Suc } n) / \text{of-nat } (\text{Suc } n) \text{ powr } s) \text{ sums inverse } (\text{zeta } s)$   
 proof –  
 have fds-converges (fds moebius-mu)  $s$   
 using assms by (auto intro!: fds-abs-converges-moebius-mu)  
 hence  $(\lambda n. \text{moebius-mu } (\text{Suc } n) / \text{of-nat } (\text{Suc } n) \text{ powr } s) \text{ sums eval-fds } (\text{fds moebius-mu}) s$   
 by (simp add: fds-converges-altdef)  
 also have  $\text{fds moebius-mu} = \text{inverse } (\text{fds-zeta} :: \text{complex fds})$   
 by (rule fds-moebius-inverse-zeta)  
 also from  $s$  have  $\text{eval-fds} \dots s = \text{inverse } (\text{zeta } s)$   
 by (subst eval-fds-inverse)  
 (auto simp: fds-moebius-inverse-zeta [symmetric] eval-fds-zeta  
 intro!: fds-abs-converges-moebius-mu)  
 finally show ?thesis .  
 qed

The following gives an extension of the  $\zeta$  functions to the critical strip.

**lemma** hurwitz-zeta-critical-strip:

fixes  $s :: \text{complex}$  and  $a :: \text{real}$   
 defines  $S \equiv (\lambda n. \sum_{i < n} (\text{of-nat } i + a) \text{ powr } -s)$   
 defines  $I' \equiv (\lambda n. \text{of-nat } n \text{ powr } (1 - s) / (1 - s))$   
 assumes  $\text{Re } s > 0$   $s \neq 1$  and  $a > 0$   
 shows  $(\lambda n. S n - I' n) \longrightarrow \text{hurwitz-zeta } a s$   
 proof –  
 from assms have [simp]:  $s \neq 1$  by auto  
 let  $?f = \lambda x. \text{of-real } (x + a) \text{ powr } -s$   
 let  $?fs = \lambda n x. (-1) \wedge n * \text{pochhammer } s n * \text{of-real } (x + a) \text{ powr } (-s - \text{of-nat } n)$   
 have minus-commute:  $-a - b = -b - a$  for  $a b :: \text{complex}$  by (simp add: algebra-simps)  
 define  $I$  where  $I = (\lambda n. (\text{of-nat } n + a) \text{ powr } (1 - s) / (1 - s))$   
 define  $R$  where  $R = (\lambda n. \text{EM-remainder}' 1 (?fs 1) (\text{real } 0) (\text{real } n))$   
 define  $R\text{-lim}$  where  $R\text{-lim} = \text{EM-remainder } 1 (?fs 1) 0$   
 define  $C$  where  $C = -(a \text{ powr } -s / 2)$   
 define  $D$  where  $D = (\lambda n. (1/2) * (\text{of-real } (a + \text{real } n) \text{ powr } -s))$   
 define  $D'$  where  $D' = (\lambda n. \text{of-real } (a + \text{real } n) \text{ powr } -s)$   
 define  $C'$  where  $C' = a \text{ powr } (1 - s) / (1 - s)$   
 define  $C''$  where  $C'' = \text{of-real } a \text{ powr } -s$   
 {

```

fix n :: nat assume n: n > 0
  have ((λx. of-real (x + a) powr -s) has-integral (of-real (real n + a) powr
(1-s) / (1 - s) -
  of-real (0 + a) powr (1 - s) / (1 - s))) {0..real n} using n assms
  by (intro fundamental-theorem-of-calculus)
  (auto intro!: continuous-intros has-vector-derivative-real-field derivative-eq-intros
simp: complex-nonpos-Reals-iff)
hence I: ((λx. of-real (x + a) powr -s) has-integral (I n - C')) {0..n}
  by (auto simp: divide-simps C'-def I-def)
have (∑ i∈{0<..n}. ?f (real i)) - integral {real 0..real n} ?f =
  (∑ k<1. (bernoulli' (Suc k) / fact (Suc k)) *R (?fs k (real n) - ?fs k
(real 0))) + R n
  using n assms unfolding R-def
  by (intro euler-maclaurin-strong-raw-nat[where Y = {0}])
  (auto intro!: continuous-intros derivative-eq-intros has-vector-derivative-real-field
simp: pochhammer-rec' algebra-simps complex-nonpos-Reals-iff
add-eq-0-iff)
  also have (∑ k<1. (bernoulli' (Suc k) / fact (Suc k)) *R (?fs k (real n) - ?fs
k (real 0))) =
  ((n + a) powr -s - a powr -s) / 2
  by (simp add: lessThan-nat-numeral scaleR-conv-of-real numeral-2-eq-2 [symmetric])
  also have ... = C + D n by (simp add: C-def D-def field-simps)
  also have integral {real 0..real n} (λx. complex-of-real (x + a) powr -s) = I
n - C'
  using I by (simp add: has-integral-iff)
  also have (∑ i∈{0<..n}. of-real (real i + a) powr -s) =
  (∑ i=0..n. of-real (real i + a) powr -s) - of-real a powr -s
  using assms by (subst sum.head) auto
  also have (∑ i=0..n. of-real (real i + a) powr -s) = S n + of-real (real n +
a) powr -s
  unfolding S-def by (subst sum.last-plus) (auto simp: atLeast0LessThan)
  finally have C - C' + C'' - D' n + D n + R n + (I n - I' n) = S n - I' n
  by (simp add: algebra-simps S-def D'-def C''-def)
}
hence ev: eventually (λn. C - C' + C'' - D' n + D n + R n + (I n - I' n)
= S n - I' n) at-top
  by (intro eventually-mono[OF eventually-gt-at-top[of 0]]) auto

have [simp]: -1 - s = -s - 1 by simp
{
  let ?C = norm (pochhammer s 1)
  have R → R-lim unfolding R-def R-lim-def of-nat-0
  proof (subst of-int-0 [symmetric], rule tendsto-EM-remainder)
    show eventually (λx. norm (?fs 1 x) ≤ ?C * (x + a) powr (-Re s - 1))
at-top
    using eventually-ge-at-top[of 0]
    by eventually-elim (insert assms, auto simp: norm-mult norm-powr-real-powr)
  next
  fix x assume x: x ≥ real-of-int 0

```



```

    have [simp]:  $- \text{numeral } n - (x :: \text{real}) = -x - \text{numeral } n$  for  $x$   $n$  by (simp
add: algebra-simps)
    show (( $\lambda x. ?C / (-\text{Re } s) * (x + a) \text{ powr } (-\text{Re } s)$ ) has-real-derivative
 $?C * (x + a) \text{ powr } (-\text{Re } s - 1)$ ) (at  $x$  within {real-of-int 0..})
    using assms  $x$  by (auto intro!: derivative-eq-intros)
  next
    have ( $\lambda y. ?C / (-\text{Re } s) * (a + \text{real } y) \text{ powr } (-\text{Re } s)$ )  $\longrightarrow 0$ 
    by (intro tendsto-mult-right-zero tendsto-neg-powr filterlim-real-sequentially
filterlim-tendsto-add-at-top[OF tendsto-const]) (use assms in auto)
    thus convergent ( $\lambda y. ?C / (-\text{Re } s) * (\text{real } y + a) \text{ powr } (-\text{Re } s)$ )
    by (auto simp: add-ac convergent-def)
  qed (intro integrable-EM-remainder' continuous-intros, insert assms, auto simp:
add-eq-0-iff)
}
moreover have ( $\lambda n. I\ n - I'\ n$ )  $\longrightarrow 0$ 
proof -
  have ( $\lambda n. (\text{complex-of-real } (\text{real } n + a) \text{ powr } (1 - s) -$ 
of-real ( $\text{real } n$ )  $\text{ powr } (1 - s)) / (1 - s)$ )  $\longrightarrow 0 / (1 - s)$ 
  using assms(3-5) by (intro filterlim-compose[OF filterlim-real-sequentially]
tendsto-divide complex-powr-add-minus-powr-asymptotics)
auto
  thus ( $\lambda n. I\ n - I'\ n$ )  $\longrightarrow 0$  by (simp add: I-def I'-def divide-simps)
qed
ultimately have ( $\lambda n. C - C' + C'' - D'\ n + D\ n + R\ n + (I\ n - I'\ n)$ )
 $\longrightarrow C - C' + C'' - 0 + 0 + R\text{-lim} + 0$ 
  unfolding D-def D'-def using assms
  by (intro tendsto-add tendsto-diff tendsto-const tendsto-mult-right-zero
tendsto-neg-powr-complex-of-real filterlim-tendsto-add-at-top
filterlim-real-sequentially) auto
also have  $C - C' + C'' - 0 + 0 + R\text{-lim} + 0 =$ 
 $(a \text{ powr } -s / 2) + a \text{ powr } (1 - s) / (s - 1) + R\text{-lim}$ 
  by (simp add: C-def C'-def C''-def field-simps)
also have  $\dots = \text{hurwitz-zeta } a\ s$ 
  using assms by (simp add: hurwitz-zeta-def pre-zeta-def pre-zeta-aux-def
R-lim-def scaleR-conv-of-real)
  finally have ( $\lambda n. C - C' + C'' - D'\ n + D\ n + R\ n + (I\ n - I'\ n)$ )  $\longrightarrow$ 
hurwitz-zeta  $a\ s$  .
  with ev show ?thesis
  by (blast intro: Lim-transform-eventually)
qed

lemma zeta-critical-strip:
  fixes  $s :: \text{complex}$  and  $a :: \text{real}$ 
  defines  $S \equiv (\lambda n. \sum_{i=1..n.} (\text{of-nat } i) \text{ powr } -s)$ 
  defines  $I \equiv (\lambda n. \text{of-nat } n \text{ powr } (1 - s) / (1 - s))$ 
  assumes  $s: \text{Re } s > 0\ s \neq 1$ 
  shows ( $\lambda n. S\ n - I\ n$ )  $\longrightarrow \text{zeta } s$ 
proof -
  from hurwitz-zeta-critical-strip[OF  $s$  zero-less-one]

```

**have**  $(\lambda n. (\sum i < n. \text{complex-of-real } (\text{Suc } i) \text{ powr } -s) -$   
 $\text{of-nat } n \text{ powr } (1 - s) / (1 - s)) \longrightarrow \text{hurwitz-zeta } 1 \ s$  **by**  $(\text{simp add:}$   
 $\text{add-ac})$   
**also have**  $(\lambda n. (\sum i < n. \text{complex-of-real } (\text{Suc } i) \text{ powr } -s)) = (\lambda n. (\sum i = 1..n.$   
 $\text{of-nat } i \text{ powr } -s))$   
**by**  $(\text{intro ext sum.reindex-bij-witness}[\text{of } - \lambda x. x - 1 \ \text{Suc}]) \text{ auto}$   
**finally show**  $?thesis$  **by**  $(\text{simp add: zeta-def S-def I-def})$   
**qed**

## 2.4 The non-vanishing of $\zeta$ for $\Re(s) \geq 1$

This proof is based on a sketch by Newman [6], which was previously formalised in HOL Light by Harrison [5], albeit in a much more concrete and low-level style.

Our aim here is to reproduce Newman's proof idea cleanly and on the same high level of abstraction.

**theorem** *zeta-Re-ge-1-nonzero*:

**fixes**  $s$  **assumes**  $\text{Re } s \geq 1 \ s \neq 1$

**shows**  $\text{zeta } s \neq 0$

**proof**  $(\text{cases } \text{Re } s > 1)$

**case** *False*

**define**  $a$  **where**  $a = -\text{Im } s$

**from** *False* **assms** **have**  $s$   $[\text{simp}]$ :  $s = 1 - i * a$  **and**  $a$ :  $a \neq 0$

**by**  $(\text{auto simp: complex-eq-iff } a\text{-def})$

**show**  $?thesis$

**proof**

**assume**  $\text{zeta } s = 0$

**hence**  $\text{zero: zeta } (1 - i * a) = 0$  **by**  $\text{simp}$

**with**  $\text{zeta-cn}j[\text{of } 1 - i * a]$  **have**  $\text{zero': zeta } (1 + i * a) = 0$  **by**  $\text{simp}$

— We define the function  $Q(s) = \zeta(s)^2 \zeta(s+ia) \zeta(s-ia)$  and its Dirichlet series. The objective will be to show that this function is entire and its Dirichlet series converges everywhere. Of course,  $Q(s)$  has singularities at 1 and  $1 \pm ia$ , so we need to show they can be removed.

**define**  $Q$   $Q\text{-fds}$

**where**  $Q = (\lambda s. \text{zeta } s \wedge 2 * \text{zeta } (s + i * a) * \text{zeta } (s - i * a))$

**and**  $Q\text{-fds} = \text{fds-zeta } \wedge 2 * \text{fds-shift } (i * a) \text{ fds-zeta } * \text{fds-shift } (-i * a)$   
 $\text{fds-zeta}$

**let**  $?sings = \{1, 1 + i * a, 1 - i * a\}$

— We show that  $Q$  is locally bounded everywhere. This is the case because the poles of  $\zeta(s)$  cancel with the zeros of  $\zeta(s \pm ia)$  and vice versa. This boundedness is then enough to show that  $Q$  has only removable singularities.

**have**  $Q\text{-bigo-1}$ :  $Q \in O[at \ s](\lambda-. 1)$  **for**  $s$

**proof** —

**have**  $Q\text{-eq}$ :  $Q = (\lambda s. (\text{zeta } s * \text{zeta } (s + i * a)) * (\text{zeta } s * \text{zeta } (s - i * a)))$

**by**  $(\text{simp add: } Q\text{-def power2-eq-square mult-ac})$

— The singularity of  $\zeta(s)$  at 1 gets cancelled by the zero of  $\zeta(s - ia)$ :

**have** *bigo1*:  $(\lambda s. \text{zeta } s * \text{zeta } (s - i * a)) \in O[at\ 1](\lambda-. 1)$   
**if**  $\text{zeta } (1 - i * a) = 0 \ a \neq 0$  **for**  $a :: \text{real}$   
**proof** —  
**have**  $(\lambda s. \text{zeta } (s - i * a) - \text{zeta } (1 - i * a)) \in O[at\ 1](\lambda s. s - 1)$   
**using** *that*  
**by** (*intro taylor-bigo-linear holomorphic-on-imp-differentiable-at*[of - -{1 + i \* a}])  
*holomorphic-intros* (*auto simp: complex-eq-iff*)  
**hence**  $(\lambda s. \text{zeta } s * \text{zeta } (s - i * a)) \in O[at\ 1](\lambda s. 1 / (s - 1) * (s - 1))$   
**using** *that* **by** (*intro landau-o.big.mult zeta-bigo-at-1*) *simp-all*  
**also have**  $(\lambda s. 1 / (s - 1) * (s - 1)) \in \Theta[at\ 1](\lambda-. 1)$   
**by** (*intro bighetaI-cong*) (*auto simp: eventually-at-filter*)  
**finally show** *?thesis* .  
**qed**

— The analogous result for  $\zeta(s)\zeta(s + ia)$ :

**have** *bigo1'*:  $(\lambda s. \text{zeta } s * \text{zeta } (s + i * a)) \in O[at\ 1](\lambda-. 1)$   
**if**  $\text{zeta } (1 - i * a) = 0 \ a \neq 0$  **for**  $a :: \text{real}$   
**using** *bigo1*[of -a] *that zeta-cnj*[of 1 - i \* a] **by** *simp*

— The singularity of  $\zeta(s - ia)$  gets cancelled by the zero of  $\zeta(s)$ :

**have** *bigo2*:  $(\lambda s. \text{zeta } s * \text{zeta } (s - i * a)) \in O[at\ (1 + i * a)](\lambda-. 1)$   
**if**  $\text{zeta } (1 - i * a) = 0 \ a \neq 0$  **for**  $a :: \text{real}$   
**proof** —  
**have**  $(\lambda s. \text{zeta } s * \text{zeta } (s - i * a)) \in O[\text{filtermap } (\lambda s. s + i * a) (at\ 1)](\lambda-. 1)$   
**using** *bigo1'*[of a] *that* **by** (*simp add: mult.commute landau-o.big.in-filtermap-iff*)  
**also have**  $\text{filtermap } (\lambda s. s + i * a) (at\ 1) = at\ (1 + i * a)$   
**using** *filtermap-at-shift*[of -i \* a 1] **by** *simp*  
**finally show** *?thesis* .  
**qed**

— Again, the analogous result for  $\zeta(s)\zeta(s + ia)$ :

**have** *bigo2'*:  $(\lambda s. \text{zeta } s * \text{zeta } (s + i * a)) \in O[at\ (1 - i * a)](\lambda-. 1)$   
**if**  $\text{zeta } (1 - i * a) = 0 \ a \neq 0$  **for**  $a :: \text{real}$   
**using** *bigo2*[of -a] *that zeta-cnj*[of 1 - i \* a] **by** *simp*

— Now the final case distinction to show  $Q(s) \in O(1)$  for all  $s \in \mathbb{C}$ :

**consider**  $s = 1 \mid s = 1 + i * a \mid s = 1 - i * a \mid s \notin \text{?sings}$  **by** *blast*  
**thus** *?thesis*  
**proof** *cases*  
**case** 1  
**thus** *?thesis* **unfolding** *Q-eq* **using** *zero zero' a*  
**by** (*auto intro: bigo1 bigo1' landau-o.big.mult-in-1*)  
**next**  
**case** 2  
**from** a **have** *isCont*  $(\lambda s. \text{zeta } s * \text{zeta } (s + i * a)) (1 + i * a)$   
**by** (*auto intro!: continuous-intros*)

**with 2 show ?thesis unfolding Q-eq using zero zero' a**  
**by (auto intro: bigo2 landau-o.big.mult-in-1 continuous-imp-bigo-1)**  
**next**  
**case 3**  
**from a have isCont ( $\lambda s. \text{zeta } s * \text{zeta } (s - i * a)$ ) ( $1 - i * a$ )**  
**by (auto intro!: continuous-intros)**  
**with 3 show ?thesis unfolding Q-eq using zero zero' a**  
**by (auto intro: bigo2' landau-o.big.mult-in-1 continuous-imp-bigo-1)**  
**qed (auto intro!: continuous-imp-bigo-1 continuous-intros simp: Q-def complex-eq-iff)**  
**qed**

— Thus, we can remove the singularities from  $Q$  and extend it to an entire function.

**have  $\exists Q'. Q'$  holomorphic-on UNIV  $\wedge (\forall z \in \text{UNIV} - ?\text{sings}. Q' z = Q z)$**   
**by (intro removable-singularities Q-bigo-1)**  
**(auto simp: Q-def complex-eq-iff intro!: holomorphic-intros)**  
**then obtain  $Q'$  where  $Q'$ :  $Q'$  holomorphic-on UNIV  $\wedge z. z \notin ?\text{sings} \implies Q' z = Q z$  by blast**

—  $Q'$  constitutes an analytic continuation of the Dirichlet series of  $Q$ .

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

**proof –**

**have eval-fds Q-fds  $s = Q s$  using that**

**by (simp add: Q-fds-def Q-def eval-fds-mult eval-fds-power fds-abs-converges-mult**

**fds-abs-converges-power eval-fds-zeta)**

**also from that have  $\dots = Q' s$  by (subst  $Q'$ ) auto**

**finally show ?thesis .**

**qed**

— Since  $\zeta(s)$  and  $\zeta(s \pm ia)$  are completely multiplicative Dirichlet series, the logarithm of their product can be rewritten into the following nice form:

**have ln-Q-fds-eq:**

**fds-ln 0 Q-fds = fds ( $\lambda k. \text{of-real } (2 * \text{mangoldt } k / \ln k * (1 + \cos (a * \ln k)))$ )**

**proof –**

**note  $\text{simps} = \text{fds-ln-mult}[\text{where } l' = 0 \text{ and } l'' = 0] \text{fds-ln-power}[\text{where } l' = 0]$**

**fds-ln-prod[where  $l' = \lambda-. 0$ ]**

**have fds-ln 0 Q-fds =  $2 * \text{fds-ln 0 fds-zeta} + \text{fds-shift } (i * a) (\text{fds-ln 0 fds-zeta})$**

**+**

**fds-shift  $(-i * a) (\text{fds-ln 0 fds-zeta})$**

**by (auto simp: Q-fds-def simps)**

**also have completely-multiplicative-function (fds-nth (fds-zeta :: complex fds))**

**by standard auto**

**hence fds-ln (0 :: complex) fds-zeta = fds ( $\lambda n. \text{mangoldt } n /_R \ln (\text{real } n)$ )**

**by (subst fds-ln-completely-multiplicative) (auto simp: fds-eq-iff)**

**also have  $2 * \dots + \text{fds-shift } (i * a) \dots + \text{fds-shift } (-i * a) \dots =$**

$$fds (\lambda k. of-real (2 * mangoldt k / ln k * (1 + cos (a * ln k))))$$
 (is ?a = ?b)  
**proof** (intro fds-eqI, goal-cases)  
 case (1 n)  
 then consider  $n = 1 \mid n > 1$  by force  
 hence fds-nth ?a n = mangoldt n / ln (real n) \* (2 + (n powr (i \* a) + n  
 powr (-i \* a)))  
 by cases (auto simp: field-simps scaleR-conv-of-real numeral-fds)  
 also have  $n \text{ powr } (i * a) + n \text{ powr } (-i * a) = 2 * cos (of-real (a * ln n))$   
 using 1 by (subst cos-exp-eq) (simp-all add: powr-def algebra-simps)  
 also have mangoldt n / ln (real n) \* (2 + ...) =  
 $of-real (2 * mangoldt n / ln n * (1 + cos (a * ln n)))$   
 by (subst cos-of-real) simp-all  
 finally show ?case by (simp add: fds-nth-fds')  
 qed  
 finally show ?thesis .  
 qed

— It is then obvious that this logarithm series has non-negative real coefficients.

also have nonneg-dirichlet-series ...

**proof** (standard, goal-cases)

case (1 n)

from cos-ge-minus-one[of a \* ln n] have  $1 + cos (a * ln (real n)) \geq 0$  by  
 linarith

thus ?case using 1

by (cases n = 0)

(auto simp: complex-nonneg-Reals-iff fds-nth-fds' mangoldt-nonneg  
 intro!: divide-nonneg-nonneg mult-nonneg-nonneg)

qed

— Therefore, the original series also has non-negative real coefficients.

finally have nonneg: nonneg-dirichlet-series Q-fds

by (rule nonneg-dirichlet-series-lnD) (auto simp: Q-fds-def)

— By the Pringsheim–Landau theorem, a Dirichlet series with non-negative coefficients that can be analytically continued to the entire complex plane must converge everywhere, i. e. its abscissa of (absolute) convergence is  $-\infty$ :

have abscissa-Q-fds: abs-conv-abscissa Q-fds  $\leq 1$

unfolding Q-fds-def by (auto intro!: abs-conv-abscissa-mult-leI abs-conv-abscissa-power-leI)

with nonneg and eval-Q-fds and  $\langle Q' \text{ holomorphic-on UNIV} \rangle$

have abscissa: abs-conv-abscissa Q-fds =  $-\infty$

and  $g = Q \uparrow$ )  
 by (intro entire-continuation-imp-abs-conv-abscissa-MInfty[where c = 1  
 (auto simp: one-ereal-def)])

— This now leads to a contradiction in a very obvious way. If Q-fds is absolutely convergent, then the subseries corresponding to powers of 2 (i.e. we delete all summands  $a_n/n^s$  where  $n$  is not a power of 2 from the sum) is also absolutely convergent. We denote this series with R.

define R-fds where R-fds = fds-primew-subseries 2 Q-fds

have conv-abscissa R-fds  $\leq$  abs-conv-abscissa R-fds by (rule conv-le-abs-conv-abscissa)

also have *abs-conv-abscissa R-fds*  $\leq$  *abs-conv-abscissa Q-fds*  
 unfolding *R-fds-def* by (rule *abs-conv-abscissa-restrict*)  
 also have  $\dots = -\infty$  by (simp add: *abscissa*)  
 finally have *abscissa'*: *conv-abscissa R-fds*  $= -\infty$  by *simp*

— Since  $\zeta(s)$  and  $\zeta(s \pm ia)$  have an Euler product expansion for  $\Re(s) > 1$ , we have

$$R(s) = (1 - 2^{-s})^{-2} (1 - 2^{-s+ia})^{-1} (1 - 2^{-s-ia})^{-1}$$

there, and since  $R$  converges everywhere and the right-hand side is holomorphic for  $\Re(s) > 0$ , the equation is also valid for all  $s$  with  $\Re(s) > 0$  by analytic continuation.

have *eval-R*: *eval-fds R-fds*  $s =$   
 $1 / ((1 - 2^{\text{powr } -s})^2 * (1 - 2^{\text{powr } (-s + i * a)}) * (1 - 2^{\text{powr } (-s - i * a)}))$   
 (is  $- = ?f s$ ) if *Re s*  $> 0$  for  $s$   
 proof —  
 show *?thesis*  
 proof (rule *analytic-continuation-open*[where  $f = \text{eval-fds R-fds}$ ])  
 show *?f holomorphic-on*  $\{s. \text{Re } s > 0\}$   
 by (intro *holomorphic-intros*) (auto simp: *powr-def exp-eq-1 Ln-Reals-eq*)  
 next  
 fix  $z$  assume  $z: z \in \{s. \text{Re } s > 1\}$   
 have [simp]: *completely-multiplicative-function (fds-nth fds-zeta)* by *standard*  
*auto*  
 thus *eval-fds R-fds*  $z = ?f z$  using  $z$   
 by (simp add: *R-fds-def Q-fds-def eval-fds-mult eval-fds-power fds-abs-converges-mult*  
*fds-abs-converges-power fds-primew-subseries-euler-product-cm*  
*divide-simps powr-minus powr-diff powr-add fds-abs-summable-zeta*)  
 qed (insert that *abscissa'*, *auto intro!*: *exI*[of  $- 2$ ] *convex-connected open-halfspace-Re-gt*  
*convex-halfspace-Re-gt holomorphic-intros*)  
 qed

— We now clearly have a contradiction:  $R(s)$ , being entire, is continuous everywhere, while the function on the right-hand side clearly has a pole at 0.

show *False*  
 proof (rule *not-tendsto-and-filterlim-at-infinity*)  
 have  $((\lambda b. (1 - 2^{\text{powr } - b})^2 * (1 - 2^{\text{powr } (-b+i*a)}) * (1 - 2^{\text{powr } (-b-i*a)})) \longrightarrow 0)$   
 (at 0 within  $\{s. \text{Re } s > 0\}$ )  
 (is *filterlim ?f' -*) by (intro *tendsto-eq-intros*) (auto)  
 moreover have *eventually*  $(\lambda s. s \in \{s. \text{Re } s > 0\})$  (at 0 within  $\{s. \text{Re } s > 0\}$ )  
 by (auto simp: *eventually-at-filter*)  
 hence *eventually*  $(\lambda s. ?f' s \neq 0)$  (at 0 within  $\{s. \text{Re } s > 0\}$ )  
 by *eventually-elim* (auto simp: *powr-def exp-eq-1 Ln-Reals-eq*)  
 ultimately have *filterlim ?f'* (at 0) (at 0 within  $\{s. \text{Re } s > 0\}$ ) by (simp add: *filterlim-at*)  
 hence *filterlim ?f at-infinity* (at 0 within  $\{s. \text{Re } s > 0\}$ ) (is *?lim*)

```

    by (intro filterlim-divide-at-infinity[OF tendsto-const]
        tendsto-mult-filterlim-at-infinity) auto
  also have ev: eventually ( $\lambda s. \text{Re } s > 0$ ) (at 0 within  $\{s. \text{Re } s > 0\}$ )
    by (auto simp: eventually-at intro!: exI[of - 1])
  have ?lim  $\longleftrightarrow$  filterlim (eval-fds R-fds) at-infinity (at 0 within  $\{s. \text{Re } s > 0\}$ )
0})
    by (intro filterlim-cong refl eventually-mono[OF ev]) (auto simp: eval-R)
  finally show ... .
next
have continuous (at 0 within  $\{s. \text{Re } s > 0\}$ ) (eval-fds R-fds)
  by (intro continuous-intros) (auto simp: abscissa')
thus ((eval-fds R-fds  $\longrightarrow$  eval-fds R-fds 0)) (at 0 within  $\{s. \text{Re } s > 0\}$ )
  by (auto simp: continuous-within)
next
have  $0 \in \{s. \text{Re } s \geq 0\}$  by simp
also have  $\{s. \text{Re } s \geq 0\} = \text{closure } \{s. \text{Re } s > 0\}$ 
  using closure-halfspace-gt[of 1::complex 0] by (simp add: inner-commute)
finally have  $0 \in \dots$  .
thus at 0 within  $\{s. \text{Re } s > 0\} \neq \text{bot}$ 
  by (subst at-within-eq-bot-iff) auto
qed
qed
qed (fact zeta-Re-gt-1-nonzero)

```

## 2.5 Special values of the $\zeta$ functions

**theorem** hurwitz-zeta-neg-of-nat:

assumes  $a > 0$

shows  $\text{hurwitz-zeta } a \text{ } (-\text{of-nat } n) = -\text{bernpoly } (\text{Suc } n) \ a \ / \ \text{of-nat } (\text{Suc } n)$

**proof** –

have  $-\text{of-nat } n \neq (1::\text{complex})$  by (simp add: complex-eq-iff)

hence  $\text{hurwitz-zeta } a \text{ } (-\text{of-nat } n) =$

$\text{pre-zeta } a \text{ } (-\text{of-nat } n) + a \ \text{powr } \text{real } (\text{Suc } n) \ / \ (-\text{of-nat } (\text{Suc } n))$

**unfolding** zeta-def hurwitz-zeta-def **using** assms **by** (simp add: powr-of-real [symmetric])

**also have**  $a \ \text{powr } \text{real } (\text{Suc } n) \ / \ (-\text{of-nat } (\text{Suc } n)) = - \ (a \ \text{powr } \text{real } (\text{Suc } n) \ / \ \text{of-nat } (\text{Suc } n))$

**by** (simp add: divide-simps del: of-nat-Suc)

**also have**  $a \ \text{powr } \text{real } (\text{Suc } n) = a \ ^{\wedge} \ \text{Suc } n$

**using** assms **by** (intro powr-realpow)

**also have**  $\text{pre-zeta } a \text{ } (-\text{of-nat } n) = \text{pre-zeta-aux } (\text{Suc } n) \ a \ \text{ } (-\ \text{of-nat } n)$

**using** assms **by** (intro pre-zeta-aux-eq-pre-zeta [symmetric]) auto

**also have**  $\dots = \text{of-real } a \ \text{powr } \text{of-nat } n \ / \ 2 +$

$(\sum i = 1.. \text{Suc } n. (\text{bernoulli } (2 * i) \ / \ \text{fact } (2 * i)) *_{\mathbb{R}}$

$(\text{pochhammer } (-\ \text{of-nat } n) \ (2 * i - 1) *_{\mathbb{R}}$

$\text{of-real } a \ \text{powr } (\text{of-nat } n - \text{of-nat } (2 * i - 1)))) +$

$\text{EM-remainder } (\text{Suc } (2 * \text{Suc } n)) \ (\lambda x. - (\text{pochhammer } (-\ \text{of-nat } n)$

$n)$

$(2 * n + 3) * \text{of-real } (x + a) \ \text{powr } (-\ \text{of-nat } n - 3))) \ 0$

(is - = ?B + sum (λn. ?f (2 \* n)) - + -)  
**unfolding** pre-zeta-aux-def **by** (simp add: add-ac eval-nat-numeral)  
**also have** ?B = of-real (a ^ n) / 2  
**using** assms **by** (subst powr-Reals-eq) (auto simp: powr-realpow)  
**also have** pochhammer (-of-nat n :: complex) (2\*n+3) = 0  
**by** (subst pochhammer-eq-0-iff) auto  
**finally have** hurwitz-zeta a (-of-nat n) =  
 - (a ^ Suc n / of-nat (Suc n)) + (a ^ n / 2 + sum (λn. ?f (2 \*  
 n)) {1..Suc n})  
**by** simp

**also have** sum (λn. ?f (2 \* n)) {1..Suc n} = sum ?f ((\*) 2 ' {1..Suc n})  
**by** (intro sum.reindex-bij-witness[of - λi. i div 2 λi. 2\*i]) auto  
**also have** ... = (∑ i=2..2\*n+2. ?f i)  
**proof** (intro sum.mono-neutral-left ballI, goal-cases)  
**case** (3 i)  
**hence** odd i i ≠ 1 **by** (auto elim!: evenE)  
**thus** ?case **by** (simp add: bernoulli-odd-eq-0)  
**qed** auto  
**also have** ... = (∑ i=2..Suc n. ?f i)  
**proof** (intro sum.mono-neutral-right ballI, goal-cases)  
**case** (3 i)  
**hence** pochhammer (-of-nat n :: complex) (i - 1) = 0  
**by** (subst pochhammer-eq-0-iff) auto  
**thus** ?case **by** simp  
**qed** auto  
**also have** ... = (∑ i=Suc 1..Suc n. -of-real (real (Suc n choose i) \* bernoulli  
 i \*  
 a ^ (Suc n - i)) / of-nat (Suc n))  
 (is sum ?lhs - = sum ?f -)  
**proof** (intro sum.cong, goal-cases)  
**case** (2 i)  
**hence** of-nat n - of-nat (i - 1) = (of-nat (Suc n - i) :: complex)  
**by** (auto simp: of-nat-diff)  
**also have** of-real a powr ... = of-real (a ^ (Suc n - i))  
**using** 2 assms **by** (subst powr-nat) auto  
**finally have** A: of-real a powr (of-nat n - of-nat (i - 1)) = ... .  
**have** pochhammer (-of-nat n) (i - 1) = complex-of-real (pochhammer (-real  
 n) (i - 1))  
**by** (simp add: pochhammer-of-real [symmetric])  
**also have** pochhammer (-real n) (i - 1) = pochhammer (-of-nat (Suc n)) i  
 / (-1 - real n)  
**using** 2 **by** (subst (2) pochhammer-rec-if) auto  
**also have** -1 - real n = -real (Suc n) **by** simp  
**finally have** B: pochhammer (-of-nat n) (i - 1) =  
 -complex-of-real (pochhammer (-real (Suc n)) i / real (Suc n))  
**by** (simp del: of-nat-Suc)  
**have** ?lhs i = -complex-of-real (bernoulli i \* pochhammer (-real (Suc n)) i /  
 fact i \*



$a \wedge (\text{Suc } n - i) / \text{of-nat } (\text{Suc } n)$   
**by** (*simp only: A B*) (*simp add: scaleR-conv-of-real*)  
**also have**  $\text{bernoulli } i * \text{pochhammer } (-\text{real } (\text{Suc } n)) i / \text{fact } i =$   
 $(\text{real } (\text{Suc } n) \text{ gchoose } i) * \text{bernoulli } i$  **using** 2  
**by** (*subst gbinomial-pochhammer*) (*auto simp: minus-one-power-iff bernoulli-odd-eq-0*)  
**also have**  $\text{real } (\text{Suc } n) \text{ gchoose } i = \text{Suc } n \text{ choose } i$   
**by** (*subst binomial-gbinomial*) *auto*  
**finally show** ?case **by** *simp*  
**qed** *auto*  
**also have**  $a \wedge n / 2 + \text{sum } ?f \{ \text{Suc } 1 .. \text{Suc } n \} = \text{sum } ?f \{ 1 .. \text{Suc } n \}$   
**by** (*subst (2) sum.atLeast-Suc-atMost*) (*simp-all add: scaleR-conv-of-real del: of-nat-Suc*)  
**also have**  $-(a \wedge \text{Suc } n / \text{of-nat } (\text{Suc } n)) + \text{sum } ?f \{ 1 .. \text{Suc } n \} = \text{sum } ?f \{ 0 .. \text{Suc } n \}$   
**by** (*subst (2) sum.atLeast-Suc-atMost*) (*simp-all add: scaleR-conv-of-real*)  
**also have**  $\dots = -\text{bernpoly } (\text{Suc } n) a / \text{of-nat } (\text{Suc } n)$   
**unfolding** *sum-negf sum-divide-distrib [symmetric]* **by** (*simp add: bernpoly-def atLeast0AtMost*)  
**finally show** ?thesis .  
**qed**

**lemma** *hurwitz-zeta-0 [simp]:*  $a > 0 \implies \text{hurwitz-zeta } a 0 = 1 / 2 - a$   
**using** *hurwitz-zeta-neg-of-nat[of a 0]* **by** (*simp add: bernpoly-def*)

**lemma** *zeta-0 [simp]:*  $\text{zeta } 0 = -1 / 2$   
**by** (*simp add: zeta-def*)

**theorem** *zeta-neg-of-nat:*  
 $\text{zeta } (-\text{of-nat } n) = -\text{of-real } (\text{bernoulli}' (\text{Suc } n)) / \text{of-nat } (\text{Suc } n)$   
**unfolding** *zeta-def* **by** (*simp add: hurwitz-zeta-neg-of-nat bernpoly-1'*)

**corollary** *zeta-trivial-zero:*  
**assumes** *even n n ≠ 0*  
**shows**  $\text{zeta } (-\text{of-nat } n) = 0$   
**using** *zeta-neg-of-nat[of n] assms* **by** (*simp add: bernoulli'-odd-eq-0*)

**theorem** *zeta-even-nat:*  
 $\text{zeta } (2 * \text{of-nat } n) =$   
 $\text{of-real } ((-1) \wedge \text{Suc } n * \text{bernoulli } (2 * n) * (2 * \text{pi}) \wedge (2 * n) / (2 * \text{fact } (2 * n)))$   
**proof** (*cases n = 0*)  
**case** *False*  
**hence**  $(\lambda k. 1 / \text{of-nat } (\text{Suc } k) \wedge (2 * n)) \text{ sums } \text{zeta } (\text{of-nat } (2 * n))$   
**by** (*intro sums-zeta-nat*) *auto*  
**from** *sums-unique2 [OF this nat-even-power-sums-complex] False* **show** ?thesis  
**by** *simp*  
**qed** (*insert zeta-neg-of-nat[of 0], simp-all*)

**corollary** *zeta-even-numeral:*

$$\text{zeta} (\text{numeral} (\text{Num.Bit0 } n)) = \text{of-real} \\
(((-1) \wedge \text{Suc} (\text{numeral } n) * \text{bernoulli} (\text{numeral} (\text{num.Bit0 } n)) * \\
(2 * \pi) \wedge \text{numeral} (\text{num.Bit0 } n) / (2 * \text{fact} (\text{numeral} (\text{num.Bit0 } n)))) \text{ (is -} \\
= ?rhs)$$

**proof** -

**have** \*:  $(2 * \text{numeral } n :: \text{nat}) = \text{numeral} (\text{Num.Bit0 } n)$   
**by** (*subst numeral.numeral-Bit0, subst mult-2, rule refl*)  
**have**  $\text{numeral} (\text{Num.Bit0 } n) = (2 * \text{numeral } n :: \text{complex})$   
**by** (*subst numeral.numeral-Bit0, subst mult-2, rule refl*)  
**also have** ... =  $2 * \text{of-nat} (\text{numeral } n)$  **by** (*simp only: of-nat-numeral of-nat-mult*)  
**also have**  $\text{zeta} \dots = ?rhs$  **by** (*subst zeta-even-nat*) (*simp only: \**)  
**finally show** *?thesis* .

**qed**

**corollary** *zeta-neg-even-numeral* [*simp*]:  $\text{zeta} (-\text{numeral} (\text{Num.Bit0 } n)) = 0$

**proof** -

**have**  $-\text{numeral} (\text{Num.Bit0 } n) = (-\text{of-nat} (\text{numeral} (\text{Num.Bit0 } n)) :: \text{complex})$   
**by** *simp*  
**also have**  $\text{zeta} \dots = 0$   
**proof** (*rule zeta-trivial-zero*)  
**have**  $\text{numeral} (\text{Num.Bit0 } n) = (2 * \text{numeral } n :: \text{nat})$   
**by** (*subst numeral.numeral-Bit0, subst mult-2, rule refl*)  
**also have** *even* ... **by** (*rule dvd-triv-left*)  
**finally show** *even* ( $\text{numeral} (\text{Num.Bit0 } n) :: \text{nat}$ ) .

**qed** *auto*

**finally show** *?thesis* .

**qed**

**corollary** *zeta-neg-numeral*:

$$\text{zeta} (-\text{numeral } n) = \\
-\text{of-real} (\text{bernoulli}' (\text{numeral} (\text{Num.inc } n)) / \text{numeral} (\text{Num.inc } n))$$

**proof** -

**have**  $-\text{numeral } n = (-\text{of-nat} (\text{numeral } n) :: \text{complex})$   
**by** *simp*  
**also have**  $\text{zeta} \dots = -\text{of-real} (\text{bernoulli}' (\text{numeral} (\text{Num.inc } n)) / \text{numeral} \\
(\text{Num.inc } n))$   
**by** (*subst zeta-neg-of-nat*) (*simp add: numeral-inc*)  
**finally show** *?thesis* .

**qed**

**corollary** *zeta-neg1*:  $\text{zeta} (-1) = -1 / 12$

**using** *zeta-neg-of-nat[of 1]* **by** (*simp add: eval-bernoulli*)

**corollary** *zeta-neg3*:  $\text{zeta} (-3) = 1 / 120$

**by** (*simp add: zeta-neg-numeral*)

**corollary** *zeta-neg5*:  $\text{zeta} (-5) = -1 / 252$

**by** (*simp add: zeta-neg-numeral*)

**corollary zeta-2:**  $\zeta 2 = \pi^2 / 6$   
**by** (*simp add: zeta-even-numeral*)

**corollary zeta-4:**  $\zeta 4 = \pi^4 / 90$   
**by** (*simp add: zeta-even-numeral fact-num-eq-if*)

**corollary zeta-6:**  $\zeta 6 = \pi^6 / 945$   
**by** (*simp add: zeta-even-numeral fact-num-eq-if*)

**corollary zeta-8:**  $\zeta 8 = \pi^8 / 9450$   
**by** (*simp add: zeta-even-numeral fact-num-eq-if*)

## 2.6 Integral relation between $\Gamma$ and $\zeta$ function

**lemma**

**assumes**  $z: \text{Re } z > 0$  **and**  $a: a > 0$

**shows** *Gamma-hurwitz-zeta-aux-integral:*

$\text{Gamma } z / (\text{of-nat } n + \text{of-real } a) \text{ powr } z =$   
 $(\int s \in \{0 < ..\}. (s \text{ powr } (z - 1) / \text{exp } ((n+a) * s)) \partial \text{lebesgue})$

**and** *Gamma-hurwitz-zeta-aux-integrable:*

$\text{set-integrable lebesgue } \{0 < ..\} (\lambda s. s \text{ powr } (z - 1) / \text{exp } ((n+a) * s))$

**proof** –

**note** *integrable = absolutely-integrable-Gamma-integral'* [*OF z*]

**let** *?INT = set-lebesgue-integral lebesgue*  $\{0 < ..\} :: (\text{real} \Rightarrow \text{complex}) \Rightarrow \text{complex}$

**let** *?INT' = set-lebesgue-integral lebesgue*  $\{0 < ..\} :: (\text{real} \Rightarrow \text{real}) \Rightarrow \text{real}$

**have** *meas1: set-borel-measurable lebesgue*  $\{0 < ..\}$

$(\lambda x. \text{of-real } x \text{ powr } (z - 1) / \text{of-real } (\text{exp } ((n+a) * x)))$

**unfolding** *set-borel-measurable-def*

**by** (*intro measurable-completion, subst measurable-lborel2,*

*intro borel-measurable-continuous-on-indicator*) (*auto intro!: continuous-intros*)

**show** *integrable1: set-integrable lebesgue*  $\{0 < ..\}$

$(\lambda s. s \text{ powr } (z - 1) / \text{exp } ((n+a) * s))$

**using** *assms by* (*intro absolutely-integrable-Gamma-integral*) *auto*

**from** *assms have pos: 0 < real n + a by* (*simp add: add-nonneg-pos*)

**hence** *complex-of-real 0 ≠ of-real (real n + a) by* (*simp only: of-real-eq-iff*)

**also have** *complex-of-real (real n + a) = of-nat n + of-real a by* *simp*

**finally have** *nz: ... ≠ 0 by auto*

**have**  $(\lambda t. \text{complex-of-real } t \text{ powr } (z - 1) / \text{of-real } (\text{exp } t)) \text{ has-integral Gamma } z) \{0 < ..\}$

**by** (*rule Gamma-integral-complex'*) *fact+*

**hence**  $\text{Gamma } z = ?INT (\lambda t. t \text{ powr } (z - 1) / \text{of-real } (\text{exp } t))$

**using** *set-lebesgue-integral-eq-integral(2)* [*OF integrable*]

**by** (*simp add: has-integral-iff exp-of-real*)

**also have** *lebesgue = density (distr lebesgue lebesgue*  $(\lambda t. (\text{real } n+a) * t))$

$(\lambda x. \text{ennreal } (\text{real } n+a))$

**using** *lebesgue-real-scale*[*of real n + a*] *pos by auto*

**also have** *set-lebesgue-integral ...*  $\{0 < ..\} (\lambda t. \text{of-real } t \text{ powr } (z - 1) / \text{of-real}$

```

(exp t) =
  set-lebesgue-integral (distr lebesgue lebesgue ( $\lambda t. (\text{real } n + a) * t$ ))
{0<..}
  ( $\lambda t. (\text{real } n + a) * t \text{ powr } (z - 1) / \text{exp } t$ ) using integrable pos
  unfolding set-lebesgue-integral-def
  by (subst integral-density) (simp-all add: exp-of-real algebra-simps scaleR-conv-of-real
set-integrable-def)
  also have ... = ?INT ( $\lambda s. (n + a) * (\text{of-real } (n+a) * \text{of-real } s) \text{ powr } (z - 1)$ )
/
  of-real (exp ((n+a) * s))
  unfolding set-lebesgue-integral-def
proof (subst integral-distr)
  show (*) ( $\text{real } n + a \in \text{lebesgue} \rightarrow_M \text{lebesgue}$ )
  using lebesgue-measurable-scaling[of real n + a, where ?'a = real]
  unfolding real-scaleR-def .
next
  have ( $\lambda x. (n+a) * (\text{indicator } \{0<..\} x *_{\mathbb{R}} (\text{of-real } x \text{ powr } (z - 1) / \text{of-real}$ 
(exp x)))
     $\in \text{lebesgue} \rightarrow_M \text{borel}$ 
  using integrable unfolding set-integrable-def by (intro borel-measurable-times)
simp-all
  thus ( $\lambda x. \text{indicator } \{0<..\} x *_{\mathbb{R}}$ 
    (complex-of-real (real n + a) * complex-of-real x powr (z - 1) / exp x))
     $\in \text{borel-measurable lebesgue}$  by simp
qed (intro Bochner-Integration.integral-cong refl, insert pos,
auto simp: indicator-def zero-less-mult-iff)
  also have ... = ?INT ( $\lambda s. ((n+a) \text{ powr } z) * (s \text{ powr } (z - 1) / \text{exp } ((n+a) *$ 
s))) using pos
  by (intro set-lebesgue-integral-cong refl allI impI, simp, subst powr-times-real)
    (auto simp: powr-diff)
  also have ... =  $(n + a) \text{ powr } z * ?INT (\lambda s. s \text{ powr } (z - 1) / \text{exp } ((n+a) * s))$ 
  unfolding set-lebesgue-integral-def
  by (subst integral-mult-right-zero [symmetric]) simp-all
  finally show  $\Gamma z / (\text{of-nat } n + \text{of-real } a) \text{ powr } z =$ 
    ?INT ( $\lambda s. s \text{ powr } (z - 1) / \text{exp } ((n+a) * s)$ )
  using nz by (auto simp add: field-simps)
qed

```

**lemma**

```

assumes x: x > 0 and a > 0
shows Gamma-hurwitz-zeta-aux-integral-real:
   $\Gamma x / (\text{real } n + a) \text{ powr } x =$ 
    set-lebesgue-integral lebesgue {0<..}
    ( $\lambda s. s \text{ powr } (x - 1) / \text{exp } ((\text{real } n + a) * s)$ )
  and Gamma-hurwitz-zeta-aux-integrable-real:
    set-integrable lebesgue {0<..} ( $\lambda s. s \text{ powr } (x - 1) / \text{exp } ((\text{real } n + a) *$ 
s))
proof -
  show set-integrable lebesgue {0<..} ( $\lambda s. s \text{ powr } (x - 1) / \text{exp } ((\text{real } n + a) * s)$ )

```

```

using absolutely-integrable-Gamma-integral[of of-real x real n + a]
unfolding set-integrable-def
proof (rule Bochner-Integration.integrable-bound, goal-cases)
case 3
have set-integrable lebesgue {0<..} ( $\lambda xa. \text{complex-of-real } xa \text{ powr } (\text{of-real } x - 1) /$ 
 $\text{of-real } (\text{exp } ((n + a) * xa)))$ )
using assms by (intro Gamma-hurwitz-zeta-aux-integrable) auto
also have ?this  $\longleftrightarrow$  integrable lebesgue
 $(\lambda s. \text{complex-of-real } (\text{indicator } \{0<..\} s *_R (s \text{ powr } (x - 1) / (\text{exp } ((n+a) * s))))))$ )
unfolding set-integrable-def
by (intro Bochner-Integration.integrable-cong refl) (auto simp: powr-Reals-eq indicator-def)
finally have set-integrable lebesgue {0<..} ( $\lambda s. s \text{ powr } (x - 1) / \text{exp } ((n+a) * s)$ )
unfolding set-integrable-def complex-of-real-integrable-eq .
thus ?case
by (simp add: set-integrable-def)
qed (insert assms, auto intro!: AE-I2 simp: indicator-def norm-divide norm-powr-real-powr)
from Gamma-hurwitz-zeta-aux-integral[of of-real x a n] and assms
have of-real ( $\text{Gamma } x / (\text{real } n + a) \text{ powr } x = \text{set-lebesgue-integral lebesgue } \{0<..\}$ 
 $(\lambda s. \text{complex-of-real } s \text{ powr } (\text{of-real } x - 1) / \text{of-real } (\text{exp } ((n+a) * s)))$ )
(is - = ?I)
by (auto simp: Gamma-complex-of-real powr-Reals-eq)
also have ?I = lebesgue-integral lebesgue
 $(\lambda s. \text{of-real } (\text{indicator } \{0<..\} s *_R (s \text{ powr } (x - 1) / \text{exp } ((n+a) * s))))$ )
unfolding set-lebesgue-integral-def
using assms by (intro Bochner-Integration.integral-cong refl)
 $(\text{auto simp: indicator-def powr-Reals-eq})$ 
also have  $\dots = \text{of-real } (\text{set-lebesgue-integral lebesgue } \{0<..\}$ 
 $(\lambda s. s \text{ powr } (x - 1) / \text{exp } ((n+a) * s)))$ )
unfolding set-lebesgue-integral-def
by (rule Bochner-Integration.integral-complex-of-real)
finally show  $\text{Gamma } x / (\text{real } n + a) \text{ powr } x = \text{set-lebesgue-integral lebesgue } \{0<..\}$ 
 $(\lambda s. s \text{ powr } (x - 1) / \text{exp } ((\text{real } n + a) * s))$ )
by (subst (asm) of-real-eq-iff)
qed

```

### theorem

```

assumes Re z > 1 and a > (0::real)
shows Gamma-times-hurwitz-zeta-integral: Gamma z * hurwitz-zeta a z =
 $(\int x \in \{0<..\}. (\text{of-real } x \text{ powr } (z - 1) * \text{of-real } (\text{exp } (-a*x) / (1 - \text{exp } (-x)))) \partial \text{lebesgue})$ 
and Gamma-times-hurwitz-zeta-integrable:
 $\text{set-integrable lebesgue } \{0<..\}$ 

```

$(\lambda x. \text{of-real } x \text{ powr } (z - 1) * \text{of-real } (\exp (-a*x) / (1 - \exp (-x))))$

**proof** –

**from** *assms* **have**  $z: \text{Re } z > 0$  **by** *simp*

**let**  $?INT = \text{set-lebesgue-integral lebesgue } \{0 < ..\} :: (\text{real} \Rightarrow \text{complex}) \Rightarrow \text{complex}$

**let**  $?INT' = \text{set-lebesgue-integral lebesgue } \{0 < ..\} :: (\text{real} \Rightarrow \text{real}) \Rightarrow \text{real}$

**have** 1: *complex-set-integrable lebesgue*  $\{0 < ..\}$   
 $(\lambda x. \text{of-real } x \text{ powr } (z - 1) / \text{of-real } (\exp ((\text{real } n + a) * x)))$  **for**  $n$   
**by** (*rule Gamma-hurwitz-zeta-aux-integrable*) (*use assms in simp-all*)

**have** 2: *summable*  $(\lambda n. \text{norm } (\text{indicator } \{0 < ..\} s *_{\mathbb{R}} (\text{of-real } s \text{ powr } (z - 1) / \text{of-real } (\exp ((n + a) * s))))$  **for**  $s$

**proof** (*cases*  $s > 0$ )

**case** *True*

**hence** *summable*  $(\lambda n. \text{norm } (\text{of-real } s \text{ powr } (z - 1)) * \exp (-a * s) * \exp (-s)^n)$

**using** *assms* **by** (*intro summable-mult summable-geometric*) *simp-all*

**with** *True* **show** *?thesis*

**by** (*simp add: norm-mult norm-divide exp-add exp-diff exp-minus field-simps exp-of-nat-mult [symmetric]*)

**qed** *simp-all*

**have** 3: *summable*  $(\lambda n. \int x. \text{norm } (\text{indicator } \{0 < ..\} x *_{\mathbb{R}} (\text{complex-of-real } x \text{ powr } (z - 1) / \text{complex-of-real } (\exp ((n + a) * x)))) \partial \text{lebesgue}$

**proof** –

**have** *summable*  $(\lambda n. \text{Gamma } (\text{Re } z) * (\text{real } n + a) \text{ powr } -\text{Re } z)$

**using** *assms* **by** (*intro summable-mult summable-hurwitz-zeta-real*) *simp-all*

**also have** *?this*  $\longleftrightarrow$  *summable*  $(\lambda n. ?INT' (\lambda s. \text{norm } (\text{of-real } s \text{ powr } (z - 1) / \text{of-real } (\exp ((n+a) * s))))$

**proof** (*intro summable-cong always-eventually allI, goal-cases*)

**case** (1  $n$ )

**have**  $\text{Gamma } (\text{Re } z) * (\text{real } n + a) \text{ powr } -\text{Re } z = \text{Gamma } (\text{Re } z) / (\text{real } n + a) \text{ powr } \text{Re } z$

**by** (*subst powr-minus*) (*simp-all add: field-simps*)

**also from** *assms* **have**  $\dots = (\int x \in \{0 < ..\}. (x \text{ powr } (\text{Re } z - 1) / \exp ((n+a) * x)) \partial \text{lebesgue})$

**by** (*subst Gamma-hurwitz-zeta-aux-integral-real*) *simp-all*

**also have**  $\dots = (\int x a \in \{0 < ..\}. \text{norm } (\text{of-real } x a \text{ powr } (z - 1) / \text{of-real } (\exp ((n+a) * x a))) \partial \text{lebesgue})$

**unfolding** *set-lebesgue-integral-def*

**by** (*intro Bochner-Integration.integral-cong refl*)  
*(auto simp: indicator-def norm-divide norm-powr-real-powr)*

**finally show** *?case* .

**qed**

**finally show** *?thesis*

**by** (*simp add: set-lebesgue-integral-def*)

**qed**

```

have sum-eq: (∑ n. indicator {0<..} s *R (of-real s powr (z - 1) / of-real (exp
((n+a) * s)))) =
      indicator {0<..} s *R (of-real s powr (z - 1) *
of-real (exp (-a * s) / (1 - exp (-s)))) for s
proof (cases s > 0)
  case True
    hence (∑ n. indicator {0..} s *R (of-real s powr (z - 1) / of-real (exp ((n+a)
* s)))) =
      (∑ n. of-real s powr (z - 1) * of-real (exp (-a * s)) * of-real (exp (-s))
^ n)
    by (intro suminf-cong)
      (auto simp: exp-add exp-minus exp-of-nat-mult [symmetric] field-simps
of-real-exp)
    also have (∑ n. of-real s powr (z - 1) * of-real (exp (-a * s)) * of-real (exp
(-s)) ^ n) =
      of-real s powr (z - 1) * of-real (exp (-a * s)) * (∑ n. of-real (exp
(-s)) ^ n)
    using True by (intro suminf-mult summable-geometric simp-all)
    also have (∑ n. complex-of-real (exp (-s)) ^ n) = 1 / (1 - of-real (exp (-s)))
    using True by (intro suminf-geometric auto)
    also have of-real s powr (z - 1) * of-real (exp (-a * s)) * ... =
      of-real s powr (z - 1) * of-real (exp (-a * s) / (1 - exp (-s)))
using ⟨a > 0⟩
  by (auto simp add: divide-simps exp-minus)
  finally show ?thesis using True by simp
qed simp-all

show set-integrable lebesgue {0<..}
  (λx. of-real x powr (z - 1) * of-real (exp (-a*x) / (1 - exp (-x))))
  using 1 unfolding sum-eq [symmetric] set-integrable-def
  by (intro integrable-suminf [OF - AE-I2] 2 3)

have (λn. ?INT (λs. s powr (z - 1) / exp ((n+a) * s))) sums lebesgue-integral
lebesgue
  (λs. ∑ n. indicator {0<..} s *R (s powr (z - 1) / exp ((n+a) * s))) (is
?A sums ?B)
  using 1 unfolding set-lebesgue-integral-def set-integrable-def
  by (rule sums-integral [OF - AE-I2 [OF 2] 3])
  also have ?A = (λn. Gamma z * (n + a) powr -z)
  using assms by (subst Gamma-hurwitz-zeta-aux-integral [symmetric])
  (simp-all add: powr-minus divide-simps)
  also have ?B = ?INT (λs. of-real s powr (z - 1) * of-real (exp (-a * s) / (1
- exp (-s))))
  unfolding sum-eq set-lebesgue-integral-def ..
  finally have (λn. Gamma z * (of-nat n + of-real a) powr -z) sums
?INT (λx. of-real x powr (z - 1) * of-real (exp (-a * x) / (1 - exp
(-x))))
  by simp
  moreover have (λn. Gamma z * (of-nat n + of-real a) powr -z) sums (Gamma

```

$z * \text{hurwitz-zeta } a z$   
**using** *assms* **by** (*intro sums-mult sums-hurwitz-zeta*) *simp-all*  
**ultimately show**  $\text{Gamma } z * \text{hurwitz-zeta } a z =$   
 $(\int x \in \{0 < ..\}. (\text{of-real } x \text{ powr } (z - 1) * \text{of-real } (\exp (-a * x) / (1 - \exp (-x)))) \partial \text{lebesgue})$   
**by** (*rule sums-unique2 [symmetric]*)  
**qed**

**corollary**

**assumes**  $\text{Re } z > 1$   
**shows** *Gamma-times-zeta-integral*:  $\text{Gamma } z * \text{zeta } z =$   
 $(\int x \in \{0 < ..\}. (\text{of-real } x \text{ powr } (z - 1) / \text{of-real } (\exp x - 1)) \partial \text{lebesgue})$   
(is *?th1*)  
**and** *Gamma-times-zeta-integrable*:  
*set-integrable lebesgue*  $\{0 < ..\}$   
 $(\lambda x. \text{of-real } x \text{ powr } (z - 1) / \text{of-real } (\exp x - 1))$  (is *?th2*)

**proof** –

**have** \*:  $(\lambda x. \text{indicator } \{0 < ..\} x *_R (\text{of-real } x \text{ powr } (z - 1) * \text{of-real } (\exp (-x) / (1 - \exp (-x)))) =$   
 $(\lambda x. \text{indicator } \{0 < ..\} x *_R (\text{of-real } x \text{ powr } (z - 1) / \text{of-real } (\exp x - 1)))$   
**by** (*intro ext*) (*simp add: field-simps exp-minus indicator-def*)  
**from** *Gamma-times-hurwitz-zeta-integral [OF assms zero-less-one]* **and** \*  
**show** *?th1* **by** (*simp add: zeta-def set-lebesgue-integral-def*)  
**from** *Gamma-times-hurwitz-zeta-integrable [OF assms zero-less-one]* **and** \*  
**show** *?th2* **by** (*simp add: zeta-def set-integrable-def*)  
**qed**

**corollary** *hurwitz-zeta-integral-Gamma-def*:

**assumes**  $\text{Re } z > 1 \ a > 0$   
**shows** *hurwitz-zeta*  $a z =$   
 $r\text{Gamma } z * (\int x \in \{0 < ..\}. (\text{of-real } x \text{ powr } (z - 1) * \text{of-real } (\exp (-a * x) / (1 - \exp (-x)))) \partial \text{lebesgue})$

**proof** –

**from** *assms* **have**  $\text{Gamma } z \neq 0$   
**by** (*subst Gamma-eq-zero-iff*) (*auto elim!: nonpos-Ints-cases*)  
**with** *Gamma-times-hurwitz-zeta-integral [OF assms]* **show** *?thesis*  
**by** (*simp add: rGamma-inverse-Gamma field-simps*)  
**qed**

**corollary** *zeta-integral-Gamma-def*:

**assumes**  $\text{Re } z > 1$   
**shows** *zeta*  $z =$   
 $r\text{Gamma } z * (\int x \in \{0 < ..\}. (\text{of-real } x \text{ powr } (z - 1) / \text{of-real } (\exp x - 1)) \partial \text{lebesgue})$

**proof** –

**from** *assms* **have**  $\text{Gamma } z \neq 0$   
**by** (*subst Gamma-eq-zero-iff*) (*auto elim!: nonpos-Ints-cases*)  
**with** *Gamma-times-zeta-integral [OF assms]* **show** *?thesis*



by (*simp add: rGamma-inverse-Gamma field-simps*)  
**qed**

**lemma** *Gamma-times-zeta-has-integral:*

**assumes**  $Re\ z > 1$   
**shows**  $((\lambda x. x\ powr\ (z - 1) / (of\text{-}real\ (exp\ x) - 1))\ has\text{-}integral\ (Gamma\ z * zeta\ z))\ \{0 < ..\}$   
**(is** (*?f has-integral -*) **)**

**proof** —

**have** (*?f has-integral set-lebesgue-integral lebesgue*  $\{0 < ..\}$  *?f*)  $\{0 < ..\}$

**using** *Gamma-times-zeta-integrable*[*OF assms*]

**by** (*intro has-integral-set-lebesgue*) *auto*

**also have** *set-lebesgue-integral lebesgue*  $\{0 < ..\}$   $?f = Gamma\ z * zeta\ z$

**using** *Gamma-times-zeta-integral*[*OF assms*] **by** *simp*

**finally show** *?thesis* .

**qed**

**lemma** *Gamma-times-zeta-has-integral-real:*

**fixes**  $z :: real$

**assumes**  $z > 1$

**shows**  $((\lambda x. x\ powr\ (z - 1) / (exp\ x - 1))\ has\text{-}integral\ (Gamma\ z * Re\ (zeta\ z)))\ \{0 < ..\}$

**proof** —

**from** *assms* **have**  $*: Re\ (of\text{-}real\ z) > 1$  **by** *simp*

**have**  $((\lambda x. Re\ (complex\text{-}of\text{-}real\ x\ powr\ (complex\text{-}of\text{-}real\ z - 1)) / (exp\ x - 1))\ has\text{-}integral$

$Gamma\ z * Re\ (zeta\ (complex\text{-}of\text{-}real\ z)))\ \{0 < ..\}$

**using** *has-integral-linear*[*OF Gamma-times-zeta-has-integral*[*OF \**] *bounded-linear-Re*]

**by** (*simp add: o-def Gamma-complex-of-real*)

**also have** *?this*  $\longleftrightarrow$  *?thesis*

**using** *assms* **by** (*intro has-integral-cong*) (*auto simp: powr-Reals-eq*)

**finally show** *?thesis* .

**qed**

**lemma** *Gamma-integral-real':*

**assumes**  $x > (0 :: real)$

**shows**  $((\lambda t. t\ powr\ (x - 1) / exp\ t)\ has\text{-}integral\ Gamma\ x)\ \{0 < ..\}$

**using** *Gamma-integral-real*[*OF assms*] **by** (*subst has-integral-closure* [*symmetric*])  
*auto*

## 2.7 An analytic proof of the infinitude of primes

We can now also do an analytic proof of the infinitude of primes.

**lemma** *primes-infinite-analytic: infinite*  $\{p :: nat. prime\ p\}$

**proof**

— Suppose the set of primes were finite.

**define**  $P :: nat\ set$  **where**  $P = \{p. prime\ p\}$

**assume** *fin: finite P*

— Then the Euler product form of the  $\zeta$  function ranges over a finite set, and since each factor is holomorphic in the positive real half-space, the product is, too.

```

define zeta' :: complex  $\Rightarrow$  complex
  where zeta' = ( $\lambda s$ . ( $\prod p \in P$ . inverse (1 - 1 / of-nat p powr s)))
have holo: zeta' holomorphic-on A if A  $\subseteq$  {s. Re s > 0} for A
proof -
  {
    fix p :: nat and s :: complex assume p: p  $\in$  P and s: s  $\in$  A
    from p have p': real p > 1
    by (subst of-nat-1 [symmetric], subst of-nat-less-iff) (simp add: prime-gt-Suc-0-nat
P-def)
    have norm (of-nat p powr s) = real p powr Re s
      by (simp add: norm-powr-real-powr)
    also have ... > real p powr 0 using p p' s that
      by (subst powr-less-cancel-iff) (auto simp: prime-gt-1-nat)
    finally have of-nat p powr s  $\neq$  1 using p by (auto simp: P-def)
  }
thus ?thesis by (auto simp: zeta'-def P-def intro!: holomorphic-intros)
qed

```

— Since the Euler product expansion of  $\zeta(s)$  is valid for all  $s$  with real value at least 1, and both  $\zeta(s)$  and the Euler product must be equal in the positive real half-space punctured at 1 by analytic continuation.

```

have eq: zeta s = zeta' s if Re s > 0 s  $\neq$  1 for s
proof (rule analytic-continuation-open[of {s. Re s > 1} {s. Re s > 0} - {1}
zeta zeta'])
  fix s assume s: s  $\in$  {s. Re s > 1}
  let ?f = ( $\lambda n$ .  $\prod p \leq n$ . if prime p then inverse (1 - 1 / of-nat p powr s) else 1)
  have eventually ( $\lambda n$ . ?f n = zeta' s) sequentially
    using eventually-ge-at-top[of Max P]
proof eventually-elim
  case (elim n)
  have P  $\neq$  {} by (auto simp: P-def intro!: exI[of - 2])
  with elim have P  $\subseteq$  {..n} using fin by auto
  thus ?case unfolding zeta'-def
    by (intro prod.mono-neutral-cong-right) (auto simp: P-def)
qed
moreover from s have ?f  $\longrightarrow$  zeta s by (intro euler-product-zeta) auto
ultimately have ( $\lambda$ -. zeta' s)  $\longrightarrow$  zeta s
  by (blast intro: Lim-transform-eventually)
thus zeta s = zeta' s by (simp add: LIMSEQ-const-iff)
qed (auto intro!: exI[of - 2] open-halfspace-Re-gt connected-open-delete con-
vex-halfspace-Re-gt
holomorphic-intros holo that intro: convex-connected)

```

— However, since the Euler product is holomorphic on the entire positive real half-space, it cannot have a pole at 1, while  $\zeta(s)$  does have a pole at 1. Since they are equal in the punctured neighbourhood of 1, this is a contradiction.

```

have ev: eventually ( $\lambda s. s \in \{s. \text{Re } s > 0\} - \{1\}$ ) (at 1)
  by (auto simp: eventually-at-filter intro!: open-halfspace-Re-gt
    eventually-mono[OF eventually-nhds-in-open[of  $\{s. \text{Re } s > 0\}$ ]])
have  $\neg$ is-pole zeta' 1
  by (rule not-is-pole-holomorphic [of  $\{s. \text{Re } s > 0\}$ ]) (auto intro!: holo open-halfspace-Re-gt)
also have is-pole zeta' 1  $\longleftrightarrow$  is-pole zeta 1 unfolding is-pole-def
  by (intro filterlim-cong refl eventually-mono [OF ev] eq [symmetric]) auto
finally show False using is-pole-zeta by contradiction
qed

```

## 2.8 The periodic zeta function

The periodic zeta function  $F(s, q)$  (as described e. g. by Apostol [1] is related to the Hurwitz zeta function. It is periodic in  $q$  with period 1 and it can be represented by a Dirichlet series that is absolutely convergent for  $\Re(s) > 1$ . If  $q \notin \mathbb{Z}$ , it furthermore convergent for  $\Re(s) > 0$ .

It is clear that for integer  $q$ , we have  $F(s, q) = F(s, 0) = \zeta(s)$ . Moreover, for non-integer  $q$ ,  $F(s, q)$  can be analytically continued to an entire function.

**definition** *fds-perzeta* :: *real*  $\Rightarrow$  *complex* *fds* **where**  
*fds-perzeta*  $q = \text{fds } (\lambda m. \exp (2 * \pi i * i * m * q))$

The definition of the periodic zeta function on the full domain is a bit unwieldy. The precise reasoning for this definition will be given later, and, in any case, it is probably more instructive to look at the derived “alternative” definitions later.

**definition** *perzeta* :: *real*  $\Rightarrow$  *complex*  $\Rightarrow$  *complex* **where**  
*perzeta*  $q' s =$   
 (if  $q' \in \mathbb{Z}$  then zeta  $s$   
 else let  $q = \text{frac } q'$  in  
 if  $s = 0$  then  $i / (2 * \pi i) * (\text{pre-zeta } q 1 - \text{pre-zeta } (1 - q) 1 + \ln (1 - q) - \ln q + \pi i * i)$   
 else if  $s \in \mathbb{N}$  then *eval-fds* (*fds-perzeta*  $q$ )  $s$   
 else *complex-of-real*  $(2 * \pi i)$  *powr*  $(s - 1) * i * \text{Gamma } (1 - s) * (i \text{ powr } (-s) * \text{hurwitz-zeta } q (1 - s) - i \text{ powr } s * \text{hurwitz-zeta } (1 - q) (1 - s))$ )

**interpretation** *fds-perzeta*: *periodic-fun-simple'* *fds-perzeta*  
**by** *standard* (*simp-all add*: *fds-perzeta-def exp-add ring-distrib exp-eq-polar cis-mult [symmetric] cis-multiple-2pi*)

**interpretation** *perzeta*: *periodic-fun-simple'* *perzeta*

**proof** –

```

have [simp]:  $n + 1 \in \mathbb{Z} \longleftrightarrow n \in \mathbb{Z}$  for  $n :: \text{real}$ 
  by (simp flip: frac-eq-0-iff add: frac.plus-1)
show periodic-fun-simple' perzeta
  by standard (auto simp: fun-eq-iff perzeta-def Let-def frac.plus-1)
qed

```

**lemma** *perzeta-frac* [*simp*]:  $\text{perzeta} (\text{frac } q) = \text{perzeta } q$   
**by** (*auto simp: perzeta-def fun-eq-iff Let-def*)

**lemma** *fds-perzeta-frac* [*simp*]:  $\text{fds-perzeta} (\text{frac } q) = \text{fds-perzeta } q$   
**using** *fds-perzeta.plus-of-int*[*of frac q [q]*] **by** (*simp add: frac-def*)

**lemma** *abs-conv-abscissa-perzeta*:  $\text{abs-conv-abscissa} (\text{fds-perzeta } q) \leq 1$   
**proof** (*rule abs-conv-abscissa-leI*)  
**fix** *c* **assume** *c*: *ereal*  $c > 1$   
**hence** *summable*  $(\lambda n. n \text{ powr } -c)$   
**by** (*simp add: summable-real-powr-iff*)  
**also have** *?this*  $\longleftrightarrow \text{fds-abs-converges} (\text{fds-perzeta } q) (\text{of-real } c)$  **unfolding**  
*fds-abs-converges-def*  
**by** (*intro summable-cong eventually-mono*[*OF eventually-gt-at-top*[*of 0*]])  
*(auto simp: norm-divide norm-powr-real-powr fds-perzeta-def powr-minus*  
*field-simps)*  
**finally show**  $\exists s. s \cdot 1 = c \wedge \text{fds-abs-converges} (\text{fds-perzeta } q) s$   
**by** (*intro exI*[*of - of-real c*]) *auto*  
**qed**

**lemma** *conv-abscissa-perzeta*:  $\text{conv-abscissa} (\text{fds-perzeta } q) \leq 1$   
**by** (*rule order.trans*[*OF conv-le-abs-conv-abscissa abs-conv-abscissa-perzeta*])

**lemma** *fds-perzeta--left-0* [*simp*]:  $\text{fds-perzeta } 0 = \text{fds-zeta}$   
**by** (*simp add: fds-perzeta-def fds-zeta-def*)

**lemma** *perzeta-0-left* [*simp*]:  $\text{perzeta } 0 s = \text{zeta } s$   
**by** (*simp add: perzeta-def eval-fds-zeta*)

**lemma** *perzeta-int*:  $q \in \mathbb{Z} \implies \text{perzeta } q = \text{zeta}$   
**by** (*simp add: perzeta-def fun-eq-iff*)

**lemma** *fds-perzeta-int*:  $q \in \mathbb{Z} \implies \text{fds-perzeta } q = \text{fds-zeta}$   
**by** (*auto simp: fds-perzeta.of-int elim!: Ints-cases*)

**lemma** *sums-fds-perzeta*:  
**assumes**  $\text{Re } s > 1$   
**shows**  $(\lambda m. \text{exp } (2 * \text{pi} * \text{i} * \text{Suc } m * q) / \text{of-nat } (\text{Suc } m) \text{ powr } s) \text{ sums}$   
 $\text{eval-fds } (\text{fds-perzeta } q) s$

**proof** –  
**have**  $\text{conv-abscissa} (\text{fds-perzeta } q) \leq 1$  **by** (*rule conv-abscissa-perzeta*)  
**also have**  $\dots < \text{ereal } (\text{Re } s)$  **using** *assms* **by** (*simp add: one-ereal-def*)  
**finally have**  $\text{fds-converges} (\text{fds-perzeta } q) s$  **by** (*intro fds-converges*) *auto*  
**hence**  $(\lambda n. \text{fds-nth} (\text{fds-perzeta } q) (\text{Suc } n) / \text{nat-power } (\text{Suc } n) s) \text{ sums}$   
 $\text{eval-fds } (\text{fds-perzeta } q) s$  **by** (*subst sums-Suc-iff*) (*auto simp: fds-converges-iff*)  
**thus** *?thesis* **by** (*simp add: fds-perzeta-def*)  
**qed**

**lemma** *sum-tendsto-fds-perzeta*:

**assumes**  $Re\ s > 1$

**shows**  $(\lambda n. \sum_{k \in \{0 <..n\}}. \exp (2 * \text{real } k * \text{pi} * q * i) * \text{of-nat } k \text{ powr } - s) \longrightarrow \text{eval-fds } (\text{fds-perzeta } q) s$

**proof** –

**have**  $(\lambda m. \exp (2 * \text{pi} * i * \text{Suc } m * q) / \text{of-nat } (\text{Suc } m) \text{ powr } s) \text{ sums } \text{eval-fds } (\text{fds-perzeta } q) s$

**by** (*intro sums-fds-perzeta assms*)

**hence**  $(\lambda n. \sum_{k < n}. \exp (2 * \text{real } (\text{Suc } k) * \text{pi} * q * i) * \text{of-nat } (\text{Suc } k) \text{ powr } -s) \longrightarrow$

$\text{eval-fds } (\text{fds-perzeta } q) s$

(*is filterlim ?f - -*) **by** (*simp add: sums-def powr-minus field-simps*)

**also have**  $?f = (\lambda n. \sum_{k \in \{0 <..n\}}. \exp (2 * \text{real } k * \text{pi} * q * i) * \text{of-nat } k \text{ powr } -s)$

**by** (*intro ext sum.reindex-bij-betw sum.reindex-bij-witness*[*of -  $\lambda n. n - 1$  Suc*]) *auto*

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

**qed**

Using the geometric series, it is easy to see that the Dirichlet series for  $F(s, q)$  has bounded partial sums for non-integer  $q$ , so it must converge for any  $s$  with  $\Re(s) > 0$ .

**lemma** *conv-abscissa-perzeta'*:

**assumes**  $q \notin \mathbb{Z}$

**shows**  $\text{conv-abscissa } (\text{fds-perzeta } q) \leq 0$

**proof** (*rule conv-abscissa-leI*)

**fix**  $c :: \text{real}$  **assume**  $c: \text{ereal } c > 0$

**have**  $\text{fds-converges } (\text{fds-perzeta } q) (\text{of-real } c)$

**proof** (*rule bounded-partial-sums-imp-fps-converges*)

**define**  $\omega$  **where**  $\omega = \exp (2 * \text{pi} * i * q)$

**have** [*simp*]:  $\text{norm } \omega = 1$  **by** (*simp add:  $\omega$ -def*)

**define**  $M$  **where**  $M = 2 / \text{norm } (1 - \omega)$

**from**  $\langle q \notin \mathbb{Z} \rangle$  **have**  $\omega \neq 1$

**by** (*auto simp:  $\omega$ -def exp-eq-1*)

**hence**  $M > 0$  **by** (*simp add:  $M$ -def*)

**show**  $Bseq (\lambda n. \sum_{k \leq n}. \text{fds-nth } (\text{fds-perzeta } q) k / \text{nat-power } k 0)$

**unfolding** *Bseq-def*

**proof** (*rule exI, safe*)

**fix**  $n :: \text{nat}$

**show**  $\text{norm } (\sum_{k \leq n}. \text{fds-nth } (\text{fds-perzeta } q) k / \text{nat-power } k 0) \leq M$

**proof** (*cases  $n = 0$* )

**case** *False*

**have**  $(\sum_{k \leq n}. \text{fds-nth } (\text{fds-perzeta } q) k / \text{nat-power } k 0) =$

$(\sum_{k \in \{1..1 + (n - 1)\}}. \omega ^ k)$  **using** *False*

**by** (*intro sum.mono-neutral-cong-right*)

(*auto simp: fds-perzeta-def fds-nth-fds exp-of-nat-mult* [*symmetric*] *mult-ac*

*$\omega$ -def*)

**also have**  $\dots = \omega * (1 - \omega ^ n) / (1 - \omega)$  **using**  $\langle \omega \neq 1 \rangle$  **and** *False*

```

    by (subst sum-gp-offset) simp
  also have  $\text{norm } \dots \leq 1 * (\text{norm } (1::\text{complex}) + \text{norm } (\omega \wedge n)) / \text{norm } (1 - \omega)$ 
    unfolding norm-mult norm-divide
    by (intro mult-mono divide-right-mono norm-triangle-ineq4) auto
  also have  $\dots = M$  by (simp add: M-def norm-power)
  finally show ?thesis .
qed (use ⟨M > 0⟩ in simp-all)
qed fact+
qed (insert c, auto)
thus  $\exists s. s \cdot 1 = c \wedge \text{fds-converges } (\text{fds-perzeta } q) s$ 
  by (intro exI[of - of-real c]) auto
qed

```

```

lemma fds-perzeta-one-half:  $\text{fds-perzeta } (1 / 2) = \text{fds } (\lambda n. (-1) \wedge n)$ 
  using Complex.DeMoivre[of pi]
  by (intro fds-eqI) (auto simp: fds-perzeta-def exp-eq-polar mult-ac)

```

```

lemma perzeta-one-half-1 [simp]:  $\text{perzeta } (1 / 2) 1 = -\ln 2$ 
proof (rule sums-unique2)
  have *:  $(1 / 2 :: \text{real}) \notin \mathbb{Z}$ 
    using fraction-not-in-ints[of 2 1] by auto
  have  $\text{fds-converges } (\text{fds-perzeta } (1 / 2)) 1$ 
    by (rule fds-converges, rule le-less-trans, rule conv-abscissa-perzeta') (use * in auto)
  hence  $(\lambda n. (-1) \wedge \text{Suc } n / \text{Suc } n) \text{ sums eval-fds } (\text{fds-perzeta } (1 / 2)) 1$ 
    unfolding fds-converges-altdef by (simp add: fds-perzeta-one-half)
  also from * have  $\text{eval-fds } (\text{fds-perzeta } (1 / 2)) 1 = \text{perzeta } (1 / 2) 1$ 
    by (simp add: perzeta-def)
  finally show  $(\lambda n. -\text{complex-of-real } ((-1) \wedge n / \text{Suc } n)) \text{ sums perzeta } (1 / 2) 1$ 
    by simp
  hence  $(\lambda n. -\text{complex-of-real } ((-1) \wedge n / \text{Suc } n)) \text{ sums -of-real } (\ln 2)$ 
    by (intro sums-minus sums-of-real alternating-harmonic-series-sums)
  thus  $(\lambda n. -\text{complex-of-real } ((-1) \wedge n / \text{Suc } n)) \text{ sums } -(\ln 2)$ 
    by (simp flip: Ln-of-real)
qed

```

## 2.9 Hurwitz's formula

We now move on to prove Hurwitz's formula relating the Hurwitz zeta function and the periodic zeta function. We mostly follow Apostol's proof, although we do make some small changes in order to make the proof more amenable to Isabelle's complex analysis library.

The big difference is that Apostol integrates along a circle with a slit, where the two sides of the slit lie on different branches of the integrand. This makes sense when looking at the integrand as a Riemann surface, but we do not have a notion of Riemann surfaces in Isabelle.

It is therefore much easier to simply cut the circle into an upper and a lower

half. In fact, the integral on the lower half can be reduced to the one on the upper half easily by symmetry, so we really only need to handle the integral on the upper half. The integration contour that we will use is therefore a semi-annulus in the upper half of the complex plane, centred around the origin.

Now, first of all, we prove the existence of an important improper integral that we will need later.

**lemma** *set-integrable-bigo*:

**fixes**  $f g :: \text{real} \Rightarrow 'a :: \{\text{banach}, \text{real-normed-field}, \text{second-countable-topology}\}$

**assumes**  $f \in O(\lambda x. g x)$  **and** *set-integrable lborel*  $\{a..\}$   $g$

**assumes**  $\bigwedge b. b \geq a \Rightarrow \text{set-integrable lborel } \{a..<b\} f$

**assumes** [*measurable*]: *set-borel-measurable borel*  $\{a..\}$   $f$

**shows** *set-integrable lborel*  $\{a..\}$   $f$

**proof** –

**from** *assms(1)* **obtain**  $C x0$  **where**  $C: C > 0 \bigwedge x. x \geq x0 \Rightarrow \text{norm } (f x) \leq C * \text{norm } (g x)$

**by** (*fastforce elim!: landau-o.bigE simp: eventually-at-top-linorder*)

**define**  $x0'$  **where**  $x0' = \max a x0$

**have** *set-integrable lborel*  $\{a..<x0'\}$   $f$

**by** (*intro assms*) (*auto simp: x0'-def*)

**moreover have** *set-integrable lborel*  $\{x0'..\}$   $f$  **unfolding** *set-integrable-def*

**proof** (*rule Bochner-Integration.integrable-bound*)

**from** *assms(2)* **have** *set-integrable lborel*  $\{x0'..\}$   $g$

**by** (*rule set-integrable-subset*) (*auto simp: x0'-def*)

**thus** *integrable lborel*  $(\lambda x. C *_R (\text{indicator } \{x0'..\} x *_R g x))$  **unfolding** *set-integrable-def*

**by** (*intro integrable-scaleR-right*) (*simp add: abs-mult norm-mult*)

**next**

**from** *assms(4)* **have** *set-borel-measurable borel*  $\{x0'..\}$   $f$

**by** (*rule set-borel-measurable-subset*) (*auto simp: x0'-def*)

**thus**  $(\lambda x. \text{indicator } \{x0'..\} x *_R f x) \in \text{borel-measurable lborel}$

**by** (*simp add: set-borel-measurable-def*)

**next**

**show** *AE x in lborel. norm*  $(\text{indicator } \{x0'..\} x *_R f x)$

$\leq \text{norm } (C *_R (\text{indicator } \{x0'..\} x *_R g x))$

**using**  $C$  **by** (*intro AE-I2*) (*auto simp: abs-mult indicator-def x0'-def*)

**qed**

**ultimately have** *set-integrable lborel*  $(\{a..<x0'\} \cup \{x0'..\}) f$

**by** (*rule set-integrable-Un*) *auto*

**also have**  $\{a..<x0'\} \cup \{x0'..\} = \{a..\}$  **by** (*auto simp: x0'-def*)

**finally show** *?thesis* .

**qed**

**lemma** *set-integrable-Gamma-hurwitz-aux2-real*:

**fixes**  $s a :: \text{real}$

**assumes**  $r > 0$  **and**  $a > 0$

**shows** *set-integrable lborel*  $\{r..\}$   $(\lambda x. x \text{ pow } s * (\exp (-a * x)) / (1 - \exp (-x)))$

```

      (is set-integrable - - ?g)
proof (rule set-integrable-bigo)
  have  $(\lambda x. \exp(-(a/2) * x))$  integrable-on  $\{r..\}$  using assms
    by (intro integrable-on-exp-minus-to-infinity) auto
  hence set-integrable lebesgue  $\{r..\}$   $(\lambda x. \exp(-(a/2) * x))$ 
    by (intro nonnegative-absolutely-integrable) simp-all
  thus set-integrable lborel  $\{r..\}$   $(\lambda x. \exp(-(a/2) * x))$ 
    by (simp add: set-integrable-def integrable-completion)
next
  fix  $y :: \text{real}$ 
  have set-integrable lborel  $\{r..y\}$  ?g using assms
    by (intro borel-integrable-atLeastAtMost') (auto intro!: continuous-intros)
  thus set-integrable lborel  $\{r..<y\}$  ?g
    by (rule set-integrable-subset) auto
next
  from assms show ?g  $\in O(\lambda x. \exp(-(a/2) * x))$ 
    by real-asymp
qed (auto simp: set-borel-measurable-def)

lemma set-integrable-Gamma-hurwitz-aux2:
  fixes  $s :: \text{complex}$  and  $a :: \text{real}$ 
  assumes  $r > 0$  and  $a > 0$ 
  shows set-integrable lborel  $\{r..\}$   $(\lambda x. x \text{ powr } -s * (\exp(-a * x)) / (1 - \exp(-x)))$ 
    (is set-integrable - - ?g)
proof -
  have set-integrable lborel  $\{r..\}$   $(\lambda x. x \text{ powr } -\text{Re } s * (\exp(-a * x)) / (1 - \exp(-x)))$ 
    (is set-integrable - - ?g')
    by (rule set-integrable-Gamma-hurwitz-aux2-real) (use assms in auto)
  also have ?this  $\longleftrightarrow$  integrable lborel  $(\lambda x. \text{indicator } \{r..\} x *_{\mathbb{R}} ?g' x)$ 
    by (simp add: set-integrable-def)
  also have  $(\lambda x. \text{indicator } \{r..\} x *_{\mathbb{R}} ?g' x) = (\lambda x. \text{norm } (\text{indicator } \{r..\} x *_{\mathbb{R}} ?g' x))$  using assms
    by (auto simp: indicator-def norm-mult norm-divide norm-powr-real-powr fun-eq-iff
      exp-of-real exp-minus norm-inverse in-Reals-norm power2-eq-square
      divide-simps)
  finally show ?thesis unfolding set-integrable-def
    by (subst (asm) integrable-norm-iff) auto
qed

lemma closed-neg-Im-slit: closed  $\{z. \text{Re } z = 0 \wedge \text{Im } z \leq 0\}$ 
proof -
  have closed  $(\{z. \text{Re } z = 0\} \cap \{z. \text{Im } z \leq 0\})$ 
    by (intro closed-Int closed-halfspace-Re-eq closed-halfspace-Im-le)
  also have  $\{z. \text{Re } z = 0\} \cap \{z. \text{Im } z \leq 0\} = \{z. \text{Re } z = 0 \wedge \text{Im } z \leq 0\}$  by blast
  finally show ?thesis .
qed

```

We define our semi-annulus path. When this path is mirrored into the



lower half of the complex plane and subtracted from the original path and the outer radius tends to  $\infty$ , this becomes a Hankel contour extending to  $-\infty$ .

**definition** *hankel-semiannulus* :: *real*  $\Rightarrow$  *nat*  $\Rightarrow$  *real*  $\Rightarrow$  *complex* **where**  
*hankel-semiannulus* *r N* = (*let* *R* = (*2* \* *N* + *1*) \* *pi* *in*  
*part-circlepath* *0 R 0 pi* +++ — Big half circle  
*linepath* (*of-real* (*-R*)) (*of-real* (*-r*)) +++ — Line on the negative real axis  
*part-circlepath* *0 r pi 0* +++ — Small half circle  
*linepath* (*of-real* *r*) (*of-real* *R*)) — Line on the positive real axis

**lemma** *path-hankel-semiannulus* [*simp*, *intro*]: *path* (*hankel-semiannulus* *r R*)  
**and** *valid-path-hankel-semiannulus* [*simp*, *intro*]: *valid-path* (*hankel-semiannulus* *r R*)  
**and** *pathfinish-hankel-semiannulus* [*simp*, *intro*]:  
*pathfinish* (*hankel-semiannulus* *r R*) = *pathstart* (*hankel-semiannulus* *r R*)  
**by** (*simp-all add: hankel-semiannulus-def Let-def*)

We set the stage for an application of the Residue Theorem. We define a function

$$f(s, z) = z^{-s} \frac{\exp(az)}{1 - \exp(-z)},$$

which will be the integrand. However, the principal branch of  $z^{-s}$  has a branch cut along the non-positive real axis, which is bad because a part of our integration path also lies on the non-positive real axis. We therefore choose a slightly different branch of  $z^{-s}$  by moving the logarithm branch along by  $90^\circ$  so that the branch cut lies on the non-positive imaginary axis instead.

**context**

**fixes** *a* :: *real*  
**fixes** *f* :: *complex*  $\Rightarrow$  *complex*  $\Rightarrow$  *complex*  
**and** *g* :: *complex*  $\Rightarrow$  *real*  $\Rightarrow$  *complex*  
**and** *h* :: *real*  $\Rightarrow$  *complex*  $\Rightarrow$  *real*  $\Rightarrow$  *complex*  
**and** *Res* :: *complex*  $\Rightarrow$  *nat*  $\Rightarrow$  *complex*  
**and** *Ln'* :: *complex*  $\Rightarrow$  *complex*  
**and** *F* :: *real*  $\Rightarrow$  *complex*  $\Rightarrow$  *complex*  
**assumes** *a*: *a*  $\in$  {*0* < .. *1*}

— Our custom branch of the logarithm  
**defines** *Ln'*  $\equiv$  ( $\lambda z. \ln (-i * z) + i * pi / 2$ )

— The integrand  
**defines** *f*  $\equiv$  ( $\lambda s z. \exp (Ln' z * (-s) + of-real a * z) / (1 - \exp z)$ )

— The integrand on the negative real axis  
**defines** *g*  $\equiv$  ( $\lambda s x. complex-of-real x powr -s * of-real (\exp (-a*x)) / of-real (1 - \exp (-x))$ )

— The integrand on the circular arcs

**defines**  $h \equiv (\lambda r s t. r * i * cis\ t * exp\ (a * (r * cis\ t) - (ln\ r + i * t) * s) / (1 - exp\ (r * cis\ t)))$

— The interesting part of the residues

**defines**  $Res \equiv (\lambda s k. exp\ (of-real\ (2 * real\ k * pi * a) * i) * of-real\ (2 * real\ k * pi)\ powr\ (-s))$

— The periodic zeta function (at least on  $\Re(s) > 1$  half-plane)

**defines**  $F \equiv (\lambda q. eval-fds\ (fds-perzeta\ q))$

**begin**

First, some basic properties of our custom branch of the logarithm:

**lemma**  $Ln'-i: Ln'\ i = i * pi / 2$

**by** (*simp add: Ln'-def*)

**lemma**  $Ln'-of-real-pos:$

**assumes**  $x > 0$

**shows**  $Ln'\ (of-real\ x) = of-real\ (ln\ x)$

**proof** —

**have**  $Ln'\ (of-real\ x) = Ln\ (of-real\ x * (-i)) + i * pi / 2$

**by** (*simp add: Ln'-def mult-ac*)

**also have**  $\dots = of-real\ (ln\ x)$  **using** *assms*

**by** (*subst Ln-times-of-real*) (*auto simp: Ln-of-real*)

**finally show** *?thesis* .

**qed**

**lemma**  $Ln'-of-real-neg:$

**assumes**  $x < 0$

**shows**  $Ln'\ (of-real\ x) = of-real\ (ln\ (-x)) + i * pi$

**proof** —

**have**  $Ln'\ (of-real\ x) = Ln\ (of-real\ (-x) * i) + i * pi / 2$

**by** (*simp add: Ln'-def mult-ac*)

**also have**  $\dots = of-real\ (ln\ (-x)) + i * pi$  **using** *assms*

**by** (*subst Ln-times-of-real*) (*auto simp: Ln-Reals-eq*)

**finally show** *?thesis* .

**qed**

**lemma**  $Ln'-times-of-real:$

$Ln'\ (of-real\ x * z) = of-real\ (ln\ x) + Ln'\ z$  **if**  $x > 0$  **z**  $\neq 0$  **for**  $z\ x$

**proof** —

**have**  $Ln'\ (of-real\ x * z) = Ln\ (of-real\ x * (-i * z)) + i * pi / 2$

**by** (*simp add: Ln'-def mult-ac*)

**also have**  $\dots = of-real\ (ln\ x) + Ln'\ z$

**using that by** (*subst Ln-times-of-real*) (*auto simp: Ln'-def Ln-of-real*)

**finally show** *?thesis* .

**qed**

**lemma**  $Ln'-cis:$

```

assumes  $t \in \{-\pi / 2 < \dots < 3 / 2 * \pi\}$ 
shows  $Ln' (cis\ t) = i * t$ 
proof -
  have  $\exp (i * \pi / 2) = i$  by (simp add: exp-eq-polar)
  hence  $Ln (- (i * cis\ t)) = i * (t - \pi / 2)$  using assms
    by (intro Ln-unique) (auto simp: algebra-simps exp-diff cis-conv-exp)
  thus ?thesis by (simp add: Ln'-def algebra-simps)
qed

```

Next, we show that the line and circle integrals are holomorphic using Leibniz's rule:

```

lemma contour-integral-part-circlepath-h:
  assumes  $r: r > 0$ 
  shows  $contour-integral (part-circlepath\ 0\ r\ 0\ \pi) (f\ s) = integral\ \{0..\pi\} (h\ r\ s)$ 
proof -
  have  $contour-integral (part-circlepath\ 0\ r\ 0\ \pi) (f\ s) =$ 
     $integral\ \{0..\pi\} (\lambda t. f\ s (r * cis\ t) * r * i * cis\ t)$ 
    by (simp add: contour-integral-part-circlepath-eq)
  also have  $integral\ \{0..\pi\} (\lambda t. f\ s (r * cis\ t) * r * i * cis\ t) = integral\ \{0..\pi\}$ 
     $(h\ r\ s)$ 
  proof (intro integral-cong)
    fix  $t$  assume  $t: t \in \{0..\pi\}$ 
    have  $-(\pi / 2) < 0$  by simp
    also have  $0 \leq t$  using  $t$  by simp
    finally have  $t \in \{-(\pi/2) < \dots < 3/2 * \pi\}$  using  $t$  by auto
    thus  $f\ s (r * cis\ t) * r * i * cis\ t = h\ r\ s\ t$ 
      using  $r$  by (simp add: f-def Ln'-times-of-real Ln'-cis h-def Ln-Reals-eq)
  qed
  finally show ?thesis .
qed

```

```

lemma integral-g-holomorphic:
  assumes  $b > 0$ 
  shows  $(\lambda s. integral\ \{b..c\} (g\ s))$  holomorphic-on A
proof -
  define  $g'$  where  $g' = (\lambda s\ t. - (of-real\ t\ powr\ (-s)) * complex-of-real (ln\ t) *$ 
     $of-real (exp (- (a * t))) / of-real (1 - exp (- t)))$ 
  have  $(\lambda s. integral (cbox\ b\ c) (g\ s))$  holomorphic-on UNIV
proof (rule leibniz-rule-holomorphic)
  fix  $s :: complex$  and  $t :: real$  assume  $t \in cbox\ b\ c$ 
  thus  $((\lambda s. g\ s\ t)$  has-field-derivative  $g'\ s\ t)$  (at s) using assms
    by (auto simp: g-def g'-def powr-def Ln-Reals-eq intro!: derivative-eq-intros)
next
  fix  $s$  show  $g\ s$  integrable-on  $cbox\ b\ c$  for  $s$  unfolding  $g$ -def using assms
    by (intro integrable-continuous continuous-intros) auto
next
  show continuous-on  $(UNIV \times cbox\ b\ c)$   $(\lambda (s, t). g'\ s\ t)$  using assms
    by (auto simp: g'-def case-prod-unfold intro!: continuous-intros)
qed auto

```

thus ?thesis by auto  
qed

lemma *integral-h-holomorphic*:

assumes  $r: r \in \{0 < .. < 2\}$

shows  $(\lambda s. \text{integral } \{b..c\} (h r s)) \text{ holomorphic-on } A$

proof –

have *no-sing*:  $\exp (r * \text{cis } t) \neq 1$  for  $t$

proof

define  $z$  where  $z = r * \text{cis } t$

assume  $\exp z = 1$

then obtain  $n$  where  $\text{norm } z = 2 * \pi * \text{of-int } |n|$

by (*auto simp: exp-eq-1 cmod-def abs-mult*)

moreover have  $\text{norm } z = r$  using  $r$  by (*simp add: z-def norm-mult*)

ultimately have *r-eq*:  $r = 2 * \pi * \text{of-int } |n|$  by *simp*

with  $r$  have  $n \neq 0$  by *auto*

moreover from  $r$  have  $r < 2 * \pi$  using *pi-gt3* by *simp*

with *r-eq* have  $|n| < 1$  by *simp*

ultimately show *False* by *simp*

qed

define  $h'$  where  $h' = (\lambda s t. \exp (a * r * \text{cis } t - (\ln r + i * t) * s) * (-\ln r - i * t) * (r * i * \text{cis } t) / (1 - \exp (r * \text{cis } t)))$

have  $(\lambda s. \text{integral } (\text{cbox } b c) (h r s)) \text{ holomorphic-on } \text{UNIV}$

proof (*rule leibniz-rule-holomorphic*)

fix  $s t$  assume  $t \in \text{cbox } b c$

thus  $(\lambda s. h r s t)$  has-field-derivative  $h' s t$  (at  $s$ ) using *no-sing r*

by (*auto intro!: derivative-eq-intros simp: h-def h'-def mult-ac Ln-Reals-eq*)

next

fix  $s$  show  $h r s$  integrable-on  $\text{cbox } b c$  using *no-sing unfolding h-def*

by (*auto intro!: integrable-continuous-real continuous-intros*)

next

show *continuous-on*  $(\text{UNIV} \times \text{cbox } b c)$   $(\lambda(s, t). h' s t)$  using *no-sing*

by (*auto simp: h'-def case-prod-unfold intro!: continuous-intros*)

qed *auto*

thus ?thesis by *auto*

qed

We now move on to the core result, which uses the Residue Theorem to relate a contour integral along a semi-annulus to a partial sum of the periodic zeta function.

lemma *hurwitz-formula-integral-semiannulus*:

fixes  $N :: \text{nat}$  and  $r :: \text{real}$  and  $s :: \text{complex}$

defines  $R \equiv \text{real } (2 * N + 1) * \pi$

assumes  $r > 0$  and  $r < 2$

shows  $\exp (-i * \pi * s) * \text{integral } \{r..R\} (\lambda x. x \text{ pow } (-s) * \exp (-a * x) / (1 - \exp (-x))) +$

$\text{integral } \{r..R\} (\lambda x. x \text{ pow } (-s) * \exp (a * x) / (1 - \exp x)) +$

$\text{contour-integral } (\text{part-circlepath } 0 R 0 \pi) (f s) +$

$$\text{contour-integral } (\text{part-circlepath } 0 \ r \ \text{pi } 0) \ (f \ s)$$

$$= -2 * \text{pi} * i * \exp(-s * \text{of-real } \text{pi} * i / 2) * (\sum_{k \in \{0 <.. N\}} \text{Res } s \ k)$$
**(is ?thesis1)**  
**and**  $f \ s$  *contour-integrable-on hankel-semiannulus*  $r \ N$   
**proof** –  
**have**  $2 < 1 * \text{pi}$  **using** *pi-gt3* **by** *simp*  
**also have**  $\dots \leq R$  **unfolding** *R-def* **by** (*intro mult-right-mono*) *auto*  
**finally have**  $R > 2$  **by** (*auto simp: R-def*)  
**note**  $r - R = \langle r > 0 \rangle \langle r < 2 \rangle$  *this*

— We integrate along the edge of a semi-annulus in the upper half of the complex plane. It consists of a big semicircle, a small semicircle, and two lines connecting the two circles, one on the positive real axis and one on the negative real axis. The integral along the big circle will vanish as the radius of the circle tends to  $\infty$ , whereas the remaining path becomes a Hankel contour, and the integral along that Hankel contour is what we are interested in, since it is connected to the Hurwitz zeta function.

**define** *big-circle* **where** *big-circle* = *part-circlepath*  $0 \ R \ 0 \ \text{pi}$   
**define** *small-circle* **where** *small-circle* = *part-circlepath*  $0 \ r \ \text{pi} \ 0$   
**define** *neg-line* **where** *neg-line* = *linepath* (*complex-of-real*  $(-R)$ ) (*complex-of-real*  $(-r)$ )  
**define** *pos-line* **where** *pos-line* = *linepath* (*complex-of-real*  $r$ ) (*complex-of-real*  $R$ )  
**define**  $\gamma$  **where**  $\gamma$  = *hankel-semiannulus*  $r \ N$   
**have**  $\gamma$ -*altdef*:  $\gamma$  = *big-circle* +++ *neg-line* +++ *small-circle* +++ *pos-line*  
**by** (*simp add: \gamma-def R-def add-ac hankel-semiannulus-def big-circle-def neg-line-def small-circle-def pos-line-def*)  
**have** [*simp*]: *path*  $\gamma$  *valid-path*  $\gamma$  *pathfinish*  $\gamma$  = *pathstart*  $\gamma$   
**by** (*simp-all add: \gamma-def*)

— The integrand has a branch cut along the non-positive imaginary axis and additional simple poles at  $2n\pi i$  for any  $n \in \mathbb{N}_{>0}$ . The radius of the smaller circle will always be less than  $2\pi$  and the radius of the bigger circle of the form  $(2N+1)\pi$ , so we always have precisely the first  $N$  poles inside our path.

**define** *sngs* **where** *sngs* = ( $\lambda n.$  *of-real*  $(2 * \text{pi} * \text{real } n) * i$ ) ‘ $\{0 <.. \}$ ’  
**define** *sngs'* **where** *sngs'* = ( $\lambda n.$  *of-real*  $(2 * \text{pi} * \text{real } n) * i$ ) ‘ $\{0 <.. N\}$ ’  
**have** *sngs-subset*: *sngs'*  $\subseteq$  *sngs* **unfolding** *sngs-def sngs'-def* **by** (*intro image-mono*) *auto*  
**have** *closed-sngs* [*intro*]: *closed* (*sngs* – *sngs'*) **unfolding** *sngs-def*  
**proof** (*rule discrete-imp-closed[of 2 \* pi]; safe?*)  
**fix**  $m \ n :: \text{nat}$   
**assume** *dist* (*of-real*  $(2 * \text{pi} * \text{real } m) * i$ ) (*of-real*  $(2 * \text{pi} * \text{real } n) * i$ )  $< 2 * \text{pi}$   
**also have** *dist* (*of-real*  $(2 * \text{pi} * \text{real } m) * i$ ) (*of-real*  $(2 * \text{pi} * \text{real } n) * i$ ) =  
*norm* (*of-real*  $(2 * \text{pi} * \text{real } m) * i$  – *of-real*  $(2 * \text{pi} * \text{real } n) * i$ )  
**by** (*simp add: dist-norm*)  
**also have** *of-real*  $(2 * \text{pi} * \text{real } m) * i$  – *of-real*  $(2 * \text{pi} * \text{real } n) * i$  =  
*of-real*  $(2 * \text{pi}) * i * \text{of-int}$  (*int*  $m$  – *int*  $n$ ) **by** (*simp add: algebra-simps*)  
**also have** *norm*  $\dots = 2 * \text{pi} * \text{of-int}$   $|\text{int } m - \text{int } n|$

**unfolding** *norm-mult norm-of-int* **by** (*simp add: norm-mult*)  
**finally have**  $|real\ m - real\ n| < 1$  **by** *simp*  
**hence**  $m = n$  **by** *linarith*  
**thus** *of-real*  $(2 * pi * real\ m) * i = of-real\ (2 * pi * real\ n) * i$  **by** *simp*  
**qed** *auto*

— We define an area within which the integrand is holomorphic. Choosing this area as big as possible makes things easier later on, so we only remove the branch cut and the poles.

**define**  $S$  **where**  $S = - \{z. Re\ z = 0 \wedge Im\ z \leq 0\} - (sngs - sngs')$   
**define**  $S'$  **where**  $S' = - \{z. Re\ z = 0 \wedge Im\ z \leq 0\}$

**have** *sngs*:  $exp\ z = 1 \iff z \in sngs$  **if**  $Re\ z \neq 0 \vee Im\ z > 0$  **for**  $z$   
**proof**

**assume**  $exp\ z = 1$   
**then obtain**  $n$  **where**  $n: z = 2 * pi * of-int\ n * i$   
**unfolding** *exp-eq-1* **by** (*auto simp: complex-eq-iff mult-ac*)  
**moreover from**  $n$  **and** *pi-gt-zero* **and that have**  $n > 0$  **by** (*auto simp: zero-less-mult-iff*)  
**ultimately have**  $z = of-real\ (2 * pi * nat\ n) * i$  **and**  $nat\ n \in \{0 < ..\}$   
**by** *auto*  
**thus**  $z \in sngs$  **unfolding** *sngs-def* **by** *blast*  
**qed** (*insert that, auto simp: sngs-def exp-eq-polar*)

— We show that the path stays within the well-behaved area.

**have** *path-image neg-line*  $= of-real\ \{-R..-r\}$  **using**  $r-R$   
**by** (*auto simp: neg-line-def closed-segment-Reals closed-segment-eq-real-ivl*)  
**hence** *path-image neg-line*  $\subseteq S - sngs'$  **using**  $r-R$  *sngs-subset*  
**by** (*auto simp: S-def sngs-def complex-eq-iff*)

**have** *path-image pos-line*  $= of-real\ \{r..R\}$  **using**  $r-R$   
**by** (*auto simp: pos-line-def closed-segment-Reals closed-segment-eq-real-ivl*)  
**hence** *path-image pos-line*  $\subseteq S - sngs'$  **using**  $r-R$  *sngs-subset*  
**by** (*auto simp: S-def sngs-def complex-eq-iff*)

**have** *part-circlepath-in-S*:  $z \in S - sngs'$   
**if**  $z \in path-image\ (part-circlepath\ 0\ r'\ 0\ pi) \vee z \in path-image\ (part-circlepath\ 0\ r'\ pi\ 0)$   
**and**  $r' > 0$   $r' \notin (\lambda n. 2 * pi * real\ n) \ \{-0 < ..\}$  **for**  $z\ r'$

**proof** —

**have**  $z: norm\ z = r' \wedge Im\ z \geq 0$  **using** *that*  
**by** (*auto simp: path-image-def part-circlepath-def norm-mult Im-exp linepath-def intro!: mult-nonneg-nonneg sin-ge-zero*)  
**hence**  $Re\ z \neq 0 \vee Im\ z > 0$  **using** *that* **by** (*auto simp: cmod-def*)  
**moreover from**  $z$  **and that have**  $z \notin sngs$   
**by** (*auto simp: sngs-def norm-mult image-iff*)  
**ultimately show**  $z \in S - sngs'$  **using** *sngs-subset* **by** (*auto simp: S-def*)  
**qed**

```

{
  fix n :: nat assume n: n > 0
  have r < 2 * pi * 1 using pi-gt3 r-R by simp
  also have ... ≤ 2 * pi * real n using n by (intro mult-left-mono) auto
  finally have r < ... .
}
hence r ∉ (λn. 2 * pi * real n) ‘ {0<..} using r-R by auto
from part-circlepath-in-S[OF - - this] r-R have path-image small-circle ⊆ S -
sngs'
  by (auto simp: small-circle-def)

{
  fix n :: nat assume n: n > 0 2 * pi * real n = real (2 * N + 1) * pi
  hence real (2 * n) = real (2 * N + 1) unfolding of-nat-mult by simp
  hence False unfolding of-nat-eq-iff by presburger
}
hence R ∉ (λn. 2 * pi * real n) ‘ {0<..} unfolding R-def by force
from part-circlepath-in-S[OF - - this] r-R have path-image big-circle ⊆ S - sngs'
  by (auto simp: big-circle-def)

note path-images =
  ⟨path-image small-circle ⊆ S - sngs'⟩ ⟨path-image big-circle ⊆ S - sngs'⟩
  ⟨path-image neg-line ⊆ S - sngs'⟩ ⟨path-image pos-line ⊆ S - sngs'⟩
have path-image γ ⊆ S - sngs' using path-images
  by (auto simp: γ-altdef path-image-join big-circle-def neg-line-def
      small-circle-def pos-line-def)

```

— We need to show that the complex plane is still connected after we removed the branch cut and the poles. We do this by showing that the complex plane with the branch cut removed is starlike and therefore connected. Then we remove the (countably many) poles, which does not break connectedness either.

```

have open S using closed-neg-Im-slit by (auto simp: S-def)
have starlike (UNIV - {z. Re z = 0 ∧ Im z ≤ 0})
  (is starlike ?S') unfolding starlike-def
proof (rule bexI ballI)+
  have 1 ≤ 1 * pi using pi-gt3 by simp
  also have ... < (2 + 2 * real N) * pi by (intro mult-strict-right-mono) auto
  finally show *: i ∈ ?S' by (auto simp: S-def)
  fix z assume z: z ∈ ?S'
  have closed-segment i z ∩ {z. Re z = 0 ∧ Im z ≤ 0} = {}
  proof safe
    fix s assume s: s ∈ closed-segment i z Re s = 0 Im s ≤ 0
    then obtain t where t: t ∈ {0..1} s = linepath i z t
      using linepath-image-01 by blast
    with z s t have z': Re z = 0 Im z > 0
      by (auto simp: Re-linepath' S-def linepath-0')
    with s have Im s ∈ closed-segment 1 (Im z) ∧ Im s ≤ 0
      by (subst (asm) closed-segment-same-Re) auto
    with z' show s ∈ {}
  end
end

```

by (auto simp: closed-segment-eq-real-ivl split: if-splits)  
 qed  
 thus closed-segment  $i z \subseteq ?S'$  by (auto simp: S-def)  
 qed  
 hence connected  $?S'$  by (rule starlike-imp-connected)  
 hence connected  $S'$  by (simp add: Compl-eq-Diff-UNIV S'-def)  
 have connected  $S$  unfolding S-def  
 by (rule connected-open-diff-countable)  
 (insert ⟨connected  $S'$ ⟩, auto simp: sngs-def closed-neg-Im-slit S'-def)

— The integrand is now clearly holomorphic on  $S - sngs'$  and we can apply the Residue Theorem.

have holo:  $f s$  holomorphic-on  $(S - sngs')$   
 unfolding f-def Ln'-def S-def using sngs  
 by (auto intro!: holomorphic-intros simp: complex-nonpos-Reals-iff)  
 have contour-integral  $\gamma (f s) =$   
 of-real  $(2 * \pi) * i * (\sum z \in sngs'. \text{winding-number } \gamma z * \text{residue } (f s) z)$   
 proof (rule Residue-theorem)  
 show  $\forall z. z \notin S \longrightarrow \text{winding-number } \gamma z = 0$   
 proof safe  
 fix  $z$  assume  $z \notin S$   
 hence  $\text{Re } z = 0 \wedge \text{Im } z \leq 0 \vee z \in sngs - sngs'$  by (auto simp: S-def)  
 thus  $\text{winding-number } \gamma z = 0$   
 proof  
 define  $x$  where  $x = -\text{Im } z$   
 assume  $\text{Re } z = 0 \wedge \text{Im } z \leq 0$   
 hence  $x: z = -\text{of-real } x * i \ x \geq 0$  unfolding complex-eq-iff by (simp-all  
 add: x-def)  
 obtain  $B$  where  $\bigwedge z. \text{norm } z \geq B \implies \text{winding-number } \gamma z = 0$   
 using winding-number-zero-at-infinity[of  $\gamma$ ] by auto  
 hence  $\text{winding-number } \gamma (-\text{of-real } (\max B 0) * i) = 0$  by (auto simp:  
 norm-mult)  
 also have  $\text{winding-number } \gamma (-\text{of-real } (\max B 0) * i) = \text{winding-number } \gamma$   
 $z$   
 proof (rule winding-number-eq)  
 from  $x$  have closed-segment  $(-\text{of-real } (\max B 0) * i) z \subseteq \{z. \text{Re } z = 0$   
 $\wedge \text{Im } z \leq 0\}$   
 by (auto simp: closed-segment-same-Re closed-segment-eq-real-ivl)  
 with ⟨path-image  $\gamma \subseteq S - sngs'$ ⟩  
 show closed-segment  $(-\text{of-real } (\max B 0) * i) z \cap \text{path-image } \gamma = \{\}$   
 by (auto simp: S-def)  
 qed auto  
 finally show  $\text{winding-number } \gamma z = 0$  .

next  
 assume  $z: z \in sngs - sngs'$   
 show  $\text{winding-number } \gamma z = 0$   
 proof (rule winding-number-zero-outside)  
 have path-image  $\gamma = \text{path-image big-circle} \cup \text{path-image neg-line} \cup$   
 $\text{path-image small-circle} \cup \text{path-image pos-line}$



**unfolding**  $\gamma$ -altdef small-circle-def big-circle-def pos-line-def neg-line-def  
**by** (simp add: path-image-join Un-assoc)  
**also have**  $\dots \subseteq \text{cball } 0 ((2 * N + 1) * \pi)$  **using**  $r$ - $R$   
**by** (auto simp: small-circle-def big-circle-def pos-line-def neg-line-def  
path-image-join norm-mult  $R$ -def path-image-part-circlepath'  
in-Reals-norm closed-segment-Reals closed-segment-eq-real-ivl)  
**finally show** path-image  $\gamma \subseteq \dots$  .  
**qed** (insert  $z$ , auto simp: sngs-def sngs'-def norm-mult)  
**qed**  
**qed**  
**qed** (insert  $\langle \text{path-image } \gamma \subseteq S - \text{sngs}' \rangle \langle \text{connected } S \rangle \langle \text{open } S \rangle$  holo, auto simp:  
sngs'-def)

— We can use Wenda Li's framework to compute the winding numbers at the poles and show that they are all 1.

**also have** winding-number  $\gamma z = 1$  **if**  $z \in \text{sngs}'$  **for**  $z$   
**proof** —  
**have**  $r < 2 * \pi * 1$  **using**  $\pi$ -gt3  $r$ - $R$  **by** simp  
**also have**  $\dots \leq 2 * \pi * \text{real } n$  **if**  $n > 0$  **for**  $n$  **using** that **by** (intro  
mult-left-mono) auto  
**finally have** norm- $z$ : norm  $z > r$  norm  $z < R$  **using** that  $r$ - $R$   
**by** (auto simp: sngs'-def norm-mult  $R$ -def)

**have** cindex-pathE big-circle  $z = -1$  **using**  $r$ - $R$  that **unfolding** big-circle-def  
**by** (subst cindex-pathE-circlepath-upper(1)) (auto simp: sngs'-def norm-mult  
 $R$ -def)

**have** cindex-pathE small-circle  $z = -1$  **using**  $r$ - $R$  that norm- $z$  **unfolding**  
small-circle-def

**by** (subst cindex-pathE-reversepath', subst reversepath-part-circlepath,  
subst cindex-pathE-circlepath-upper(2)) (auto simp: sngs'-def norm-mult)

**have** cindex-pathE neg-line  $z = 0$  cindex-pathE pos-line  $z = 0$

**unfolding** neg-line-def pos-line-def **using**  $r$ - $R$  that

**by** (subst cindex-pathE-linepath; force simp: neg-line-def cindex-pathE-linepath  
closed-segment-Reals closed-segment-eq-real-ivl sngs'-def complex-eq-iff)+

**note** indices =  $\langle \text{cindex-pathE big-circle } z = -1 \rangle \langle \text{cindex-pathE small-circle } z$   
 $= -1 \rangle$

$\langle \text{cindex-pathE neg-line } z = 0 \rangle \langle \text{cindex-pathE pos-line } z = 0 \rangle$

**show** ?thesis **unfolding**  $\gamma$ -altdef big-circle-def small-circle-def pos-line-def  
neg-line-def

**by** eval-winding (insert indices path-images that,  
auto simp: big-circle-def small-circle-def pos-line-def neg-line-def)

**qed**

**hence**  $(\sum_{z \in \text{sngs}'}. \text{winding-number } \gamma z * \text{residue } (f s) z) = (\sum_{z \in \text{sngs}'}. \text{residue}$   
 $(f s) z)$

**by** simp

**also have**  $\dots = (\sum_{k \in \{0 <..N\}}. \text{residue } (f s) (2 * \pi * \text{of-nat } k * i))$

**unfolding** sngs'-def **by** (subst sum.reindex) (auto intro!: inj-onI simp: o-def)

— Next, we compute the residues at each pole.

**also have**  $\text{residue } (f s) (2 * \pi * \text{of-nat } k * i) = -\exp(-s * \text{of-real } \pi * i / 2)$   
*\* Res s k*  
**if**  $k \in \{0 < .. N\}$  **for**  $k$  **unfolding**  $f\text{-def}$   
**proof** (*subst residue-simple-pole-deriv*)  
**show**  $\text{open } S'$  **using**  $\text{closed-neg-Im-slit}$  **by** (*auto simp: S'-def*)  
**show**  $\text{connected } S'$  **by** *fact*  
**show**  $(\lambda z. \exp(Ln' z * (-s) + \text{of-real } a * z))$   $\text{holomorphic-on } S'$   
 $(\lambda z. 1 - \exp z)$   $\text{holomorphic-on } S'$   
**by** (*auto simp: S'-def Ln'-def complex-nonpos-Reals-iff intro!: holomorphic-intros*)  
**have**  $((\lambda z. 1 - \exp z)$   $\text{has-field-derivative } -\exp(2 * \pi * k * i)$   
 $(\text{at } (\text{of-real } (2 * \pi * \text{real } k) * i))$   
**by** (*auto intro!: derivative-eq-intros*)  
**also have**  $-\exp(2 * \pi * k * i) = -1$  **by** (*simp add: exp-eq-polar*)  
**finally show**  $((\lambda z. 1 - \exp z)$   $\text{has-field-derivative } -1)$   
 $(\text{at } (\text{of-real } (2 * \pi * \text{real } k) * i))$  .  
**have**  $\text{Im } (\text{of-real } (2 * \pi * \text{real } k) * i) > 0$  **using**  $\pi\text{-gt-zero}$  **that**  
**by** *auto*  
**thus**  $\text{of-real } (2 * \pi * \text{real } k) * i \in S'$  **by** (*simp add: S'-def*)  
  
**have**  $\exp(i * \pi / 2) = i$  **by** (*simp add: exp-eq-polar*)  
**hence**  $\exp(Ln'(\text{complex-of-real } (2 * \pi * \text{real } k) * i) * -s +$   
 $\text{of-real } a * (\text{of-real } (2 * \pi * \text{real } k) * i)) / -1 =$   
 $-\exp(2 * k * a * \pi * i - s * \pi * i / 2 - s * \ln(2 * k * \pi))$  (**is**  $?R$   
 $= -$ )  
**using** *that* **by** (*subst Ln'-times-of-real*) (*simp-all add: Ln'-i algebra-simps*  
 $\text{exp-diff}$ )  
**also have**  $\dots = -\exp(-s * \text{of-real } \pi * i / 2) * \text{Res } s k$  **using** *that*  
**by** (*simp add: Res-def exp-diff powr-def exp-minus inverse-eq-divide Ln-Reals-eq*  
 $\text{mult-ac}$ )  
**finally show**  $?R = -\exp(-s * \text{of-real } \pi * i / 2) * \text{Res } s k$  .  
**qed** (*insert that, auto simp: S'-def exp-eq-polar*)  
**hence**  $(\sum k \in \{0 < .. N\}. \text{residue } (f s) (2 * \pi * \text{of-nat } k * i)) =$   
 $-\exp(-s * \text{of-real } \pi * i / 2) * (\sum k \in \{0 < .. N\}. \text{Res } s k)$   
**by** (*simp add: sum-distrib-left*)

— This gives us the final result:

**finally have**  $\text{contour-integral } \gamma (f s) =$   
 $-2 * \pi * i * \exp(-s * \text{of-real } \pi * i / 2) * (\sum k \in \{0 < .. N\}. \text{Res } s$   
 $k)$  **by** *simp*

— Lastly, we decompose the contour integral into its four constituent integrals because this makes them somewhat nicer to work with later on.

**also show**  $f s$   $\text{contour-integrable-on } \gamma$

**proof** (*rule contour-integrable-holomorphic-simple*)

**show**  $\text{path-image } \gamma \subseteq S - \text{sngs}'$  **by** *fact*

**have**  $\text{closed } \text{sngs}'$  **by** (*intro finite-imp-closed*) (*auto simp: sngs'-def*)

**with**  $\langle \text{open } S \rangle$  **show**  $\text{open } (S - \text{sngs}')$  **by** *auto*

**qed** (*insert holo, auto*)

**hence eq:**  $\text{contour-integral } \gamma (f s) =$   
 $\text{contour-integral big-circle } (f s) + \text{contour-integral neg-line } (f s) +$   
 $\text{contour-integral small-circle } (f s) + \text{contour-integral pos-line } (f s)$   
**unfolding**  $\gamma\text{-altdef big-circle-def neg-line-def small-circle-def pos-line-def}$  **by**  
 $\text{simp}$

**also have**  $\text{contour-integral neg-line } (f s) = \text{integral } \{-R..-r\} (\lambda x. f s (\text{complex-of-real } x))$   
**unfolding**  $\text{neg-line-def}$  **using**  $r\text{-}R$  **by**  $(\text{subst contour-integral-linepath-Reals-eq})$   
 $\text{auto}$   
**also have**  $\dots = \text{exp } (-i * \pi * s) * \text{integral } \{r..R\} (\lambda x. \text{exp } (-\ln x * s) * \text{exp } (-a * x) / (1 - \text{exp } (-x)))$   
**(is - = - \* ?I)** **unfolding**  $\text{integral-mult-right [symmetric]}$  **using**  $r\text{-}R$   
**by**  $(\text{subst Henstock-Kurzweil-Integration.integral-reflect-real [symmetric]})$ ,  $\text{intro integral-cong}$   
 $(\text{auto simp: f-def exp-of-real Ln'-of-real-neg exp-minus exp-Reals-eq exp-diff exp-add field-simps})$   
**also have**  $?I = \text{integral } \{r..R\} (\lambda x. x \text{powr } (-s) * \text{exp } (-a * x) / (1 - \text{exp } (-x)))$  **using**  $r\text{-}R$   
**by**  $(\text{intro integral-cong})$   $(\text{auto simp: powr-def Ln-Reals-eq exp-minus exp-diff field-simps})$

**also have**  $\text{contour-integral pos-line } (f s) = \text{integral } \{r..R\} (\lambda x. f s (\text{complex-of-real } x))$   
**unfolding**  $\text{pos-line-def}$  **using**  $r\text{-}R$  **by**  $(\text{subst contour-integral-linepath-Reals-eq})$   
 $\text{auto}$   
**also have**  $\dots = \text{integral } \{r..R\} (\lambda x. x \text{powr } (-s) * \text{exp } (a * x) / (1 - \text{exp } x))$   
**using**  $r\text{-}R$  **by**  $(\text{intro integral-cong})$   $(\text{simp add: f-def Ln'-of-real-pos exp-diff exp-minus exp-Reals-eq field-simps powr-def Ln-Reals-eq})$

**finally show**  $?thesis1$  **by**  $(\text{simp only: add-ac big-circle-def small-circle-def})$   
**qed**

Next, we need bounds on the integrands of the two semicircles.

**lemma**  $\text{hurwitz-formula-bound1}$ :

**defines**  $H \equiv \lambda z. \text{exp } (\text{complex-of-real } a * z) / (1 - \text{exp } z)$   
**assumes**  $r > 0$   
**obtains**  $C$  **where**  $C \geq 0$  **and**  $\bigwedge z. z \notin (\bigcup n::\text{int. ball } (2 * n * \pi * i) r) \implies \text{norm } (H z) \leq C$   
**proof** -  
**define**  $A$  **where**  $A = \text{cbox } (-1 - \pi * i) (1 + \pi * i) - \text{ball } 0 r$   
 $\{$   
**fix**  $z$  **assume**  $z \in A$   
**have**  $\text{exp } z \neq 1$   
**proof**  
**assume**  $\text{exp } z = 1$   
**then obtain**  $n :: \text{int}$  **where**  $[\text{simp}]: z = 2 * n * \pi * i$   
**by**  $(\text{subst (asm) exp-eq-1})$   $(\text{auto simp: complex-eq-iff})$

```

    from ⟨z ∈ A⟩ have (2 * n) * pi ≥ (-1) * pi and (2 * n) * pi ≤ 1 * pi
      by (auto simp: A-def in-cbox-complex-iff)
    hence n = 0 by (subst (asm) (1 2) mult-le-cancel-right) auto
    with ⟨z ∈ A⟩ and ⟨r > 0⟩ show False by (simp add: A-def)
  qed
}
hence continuous-on A H
  by (auto simp: A-def H-def intro!: continuous-intros)
moreover have compact A by (auto simp: A-def compact-eq-bounded-closed)
ultimately have compact (H ` A) by (rule compact-continuous-image)
hence bounded (H ` A) by (rule compact-imp-bounded)
then obtain C where bound-inside:  $\bigwedge z. z \in A \implies \text{norm } (H z) \leq C$ 
  by (auto simp: bounded-iff)

have bound-outside:  $\text{norm } (H z) \leq \exp 1 / (\exp 1 - 1)$  if  $|Re z| > 1$  for z
proof -
  have  $\text{norm } (H z) = \exp (a * Re z) / \text{norm } (1 - \exp z)$ 
    by (simp add: H-def norm-divide)
  also have  $|1 - \exp (Re z)| \leq \text{norm } (1 - \exp z)$ 
    by (rule order.trans[OF norm-triangle-ineq3]) simp
  hence  $\exp (a * Re z) / \text{norm } (1 - \exp z) \leq \exp (a * Re z) / |1 - \exp (Re z)|$ 
    using that by (intro divide-left-mono mult-pos-pos) auto
  also have  $\dots \leq \exp 1 / (\exp 1 - 1)$ 
  proof (cases Re z > 1)
    case True
      hence  $\exp (a * Re z) / |1 - \exp (Re z)| = \exp (a * Re z) / (\exp (Re z) - 1)$ 
        by simp
      also have  $\dots \leq \exp (Re z) / (\exp (Re z) - 1)$ 
        using a True by (intro divide-right-mono) auto
      also have  $\dots = 1 / (1 - \exp (-Re z))$  by (simp add: exp-minus field-simps)
      also have  $\dots \leq 1 / (1 - \exp (-1))$  using True by (intro divide-left-mono
diff-mono) auto
      also have  $\dots = \exp 1 / (\exp 1 - 1)$  by (simp add: exp-minus field-simps)
      finally show ?thesis .
    case False
      with that have  $Re z < -1$  by simp
      hence  $\exp (a * Re z) / |1 - \exp (Re z)| = \exp (a * Re z) / (1 - \exp (Re z))$ 
        by simp
      also have  $\dots \leq 1 / (1 - \exp (Re z))$ 
        using a and ⟨Re z < -1⟩ by (intro divide-right-mono) (auto intro:
mult-nonneg-nonpos)
      also have  $\dots \leq 1 / (1 - \exp (-1))$ 
        using ⟨Re z < -1⟩ by (intro divide-left-mono) auto
      also have  $\dots = \exp 1 / (\exp 1 - 1)$  by (simp add: exp-minus field-simps)
      finally show ?thesis .
  qed
  qed
finally show ?thesis .
qed

```

```

define  $D$  where  $D = \max C (exp\ 1 / (exp\ 1 - 1))$ 
have  $D \geq 0$  by (simp add: D-def max.coboundedI2)

have  $norm\ (H\ z) \leq D$  if  $z \notin (\bigcup n::int. ball\ (2 * n * pi * i)\ r)$  for  $z$ 
proof (cases |Re z| ≤ 1)
  case False
    with bound-outside[of z] show ?thesis by (simp add: D-def)
  next
    case True
      define  $n$  where  $n = \lfloor Im\ z / (2 * pi) + 1 / 2 \rfloor$ 

      have  $Im\ (z - 2 * n * pi * i) = frac\ (Im\ z / (2 * pi) + 1 / 2) * (2 * pi) - pi$ 
        by (simp add: n-def frac-def algebra-simps)
      also have  $\dots \in \{-pi..<pi\}$  using frac-lt-1 by simp
      finally have  $norm\ (H\ (z - 2 * n * pi * i)) \leq C$  using True that
        by (intro bound-inside) (auto simp: A-def in-cbox-complex-iff dist-norm n-def)
      also have  $exp\ (2 * pi * n * i) = 1$  by (simp add: exp-eq-polar)
      hence  $norm\ (H\ (z - 2 * n * pi * i)) = norm\ (H\ z)$ 
        by (simp add: H-def norm-divide exp-diff mult-ac)
      also have  $C \leq D$  by (simp add: D-def)
      finally show ?thesis .

    qed
  from  $\langle D \geq 0 \rangle$  and this show ?thesis by (rule that)
qed

lemma hurwitz-formula-bound2:
  obtains  $C$  where  $C \geq 0$  and  $\bigwedge r\ z. r > 0 \implies r < pi \implies z \in sphere\ 0\ r \implies$ 
     $norm\ (f\ s\ z) \leq C * r\ powr\ (-Re\ s - 1)$ 
proof -
  have  $2 * pi > 0$  by auto
  have  $nz: 1 - exp\ z \neq 0$  if  $z \in ball\ 0\ (2 * pi) - \{0\}$  for  $z :: complex$ 
proof
  assume  $1 - exp\ z = 0$ 
  then obtain  $n$  where  $z = 2 * pi * of-int\ n * i$ 
    by (auto simp: exp-eq-1 complex-eq-iff[of z])
  moreover have  $|real-of-int\ n| < 1 \iff n = 0$  by linarith
  ultimately show False using that by (auto simp: norm-mult)
qed

have ev: eventually  $(\lambda z::complex. 1 - exp\ z \neq 0)$  (at 0)
  using eventually-at-ball'[OF <2 * pi > 0>] by eventually-elim (use nz in auto)
have [simp]:  $subdegree\ (1 - fps-exp\ (1 :: complex)) = 1$ 
  by (intro subdegreeI) auto
hence  $(\lambda z. exp\ (a * z) * (if\ z = 0\ then\ -1\ else\ z / (1 - exp\ z :: complex)))$ 
   $has-fps-expansion\ fps-exp\ a * (fps-X / (fps-const\ 1 - fps-exp\ 1))$ 
  by (auto intro!: fps-expansion-intros)
hence  $(\lambda z::complex. exp\ (a * z) * (if\ z = 0\ then\ -1\ else\ z / (1 - exp\ z))) \in$ 
 $O[at\ 0](\lambda z. 1)$ 

```

**using** *continuous-imp-bigo-1 has-fps-expansion-imp-continuous* **by** *blast*  
**also have**  $?this \longleftrightarrow (\lambda z::\text{complex}. \exp (a * z) * (z / (1 - \exp z))) \in O[at\ 0](\lambda z.$   
 $1)$   
**by** (*intro landau-o.big.in-cong eventually-mono[OF ev]*) *auto*  
**finally have**  $\exists g. g \text{ holomorphic-on ball } 0 (2 * \pi) \wedge$   
 $(\forall z \in \text{ball } 0 (2 * \pi) - \{0\}. g z = \exp (\text{of-real } a * z) * (z / (1 -$   
 $\exp z)))$   
**using** *nz* **by** (*intro holomorphic-on-extend holomorphic-intros*) *auto*  
**then guess** *g* **by** (*elim exE conjE*) **note**  $g = this$   
**hence** *continuous-on* (*ball*  $0 (2 * \pi)$ ) *g*  
**by** (*auto dest: holomorphic-on-imp-continuous-on*)  
**hence** *continuous-on* (*cball*  $0 \pi$ ) *g*  
**by** (*rule continuous-on-subset*) (*subst cball-subset-ball-iff*, *use pi-gt-zero in auto*)  
**hence** *compact* (*g* ‘ *cball*  $0 \pi$ ) **by** (*intro compact-continuous-image*) *auto*  
**hence** *bounded* (*g* ‘ *cball*  $0 \pi$ ) **by** (*auto simp: compact-imp-bounded*)  
**then obtain** *C* **where**  $C: \forall x \in \text{cball } 0 \pi. \text{norm } (g x) \leq C$  **by** (*auto simp:*  
*bounded-iff*)

$\{$   
**fix**  $r :: \text{real}$  **assume**  $r: r > 0 \ r < \pi$   
**fix**  $z :: \text{complex}$  **assume**  $z: z \in \text{sphere } 0 \ r$   
**define**  $x$  **where**  $x = (\text{if } \text{Arg } z \leq -\pi / 2 \text{ then } \text{Arg } z + 2 * \pi \text{ else } \text{Arg } z)$   
**have**  $\exp (i * (2 * \pi)) = 1$  **by** (*simp add: exp-eq-polar*)  
**with**  $z$  **have**  $z = r * \exp (i * x)$  **using** *r pi-gt-zero Arg-eq[of z]*  
**by** (*auto simp: x-def exp-add distrib-left*)  
**have**  $x > -\pi / 2 \ x \leq 3 / 2 * \pi$  **using** *Arg-le-pi[of z] mpi-less-Arg[of z]*  
**by** (*auto simp: x-def*)  
**note**  $x = \langle z = r * \exp (i * x) \rangle this$

**from**  $x \ r$  **have**  $z': z \in \text{cball } 0 \ \pi - \{0\}$   
**using** *pi-gt3* **by** (*auto simp: norm-mult*)  
**also have**  $\text{cball } 0 \ \pi \subseteq \text{ball } (0::\text{complex}) (2 * \pi)$   
**by** (*subst cball-subset-ball-iff*) (*use pi-gt-zero in auto*)  
**hence**  $\text{cball } 0 \ \pi - \{0\} \subseteq \text{ball } 0 (2 * \pi) - \{0::\text{complex}\}$  **by** *blast*  
**finally have**  $z'': z \in \text{ball } 0 (2 * \pi) - \{0\}$ .  
**hence**  $\text{bound: norm } (\exp (a * z) * (z / (1 - \exp z))) \leq C$  **using** *C* **and** *g* **and**  
 $z'$   
**by** *force*

**have**  $\exp z \neq 1$  **using** *nz z''* **by** *auto*  
**with**  $\text{bound } z''$  **have**  $\text{bound': norm } (\exp (a * z) / (1 - \exp z)) \leq C / \text{norm } z$   
**by** (*simp add: norm-divide field-simps norm-mult*)

**have**  $\text{Ln}' z = \text{of-real } (\ln r) + \text{Ln}' (\exp (i * \text{of-real } x))$   
**using**  $x \ r$  **by** (*simp add: Ln'-times-of-real*)  
**also have**  $\exp (i * \pi / 2) = i$   
**by** (*simp add: exp-eq-polar*)  
**hence**  $\text{Ln}' (\exp (i * \text{of-real } x)) = \text{Ln} (\exp (i * \text{of-real } (x - \pi / 2))) + i * \pi /$   
 $2$

```

    by (simp add: algebra-simps Ln'-def exp-diff)
  also have ... = i * x
    using x pi-gt3 by (subst Ln-exp) (auto simp: algebra-simps)
  finally have norm (exp (-Ln' z * s)) = exp (x * Im s - ln r * Re s)
    by simp
  also {
    have x * Im s ≤ |x * Im s| by (rule abs-ge-self)
    also have ... ≤ (3/2 * pi) * |Im s| unfolding abs-mult using x
      by (intro mult-right-mono) auto
    finally have exp (x * Im s - ln r * Re s) ≤ exp (3 / 2 * pi * |Im s| - ln r
* Re s) by simp
  }
  finally have norm (exp (-Ln' z * s) * (exp (a * z) / (1 - exp z))) ≤
    exp (3 / 2 * pi * |Im s| - ln r * Re s) * (C / norm z)
    unfolding norm-mult[of exp t for t] by (intro mult-mono bound') simp-all
  also have norm z = r using ⟨r > 0⟩ by (simp add: x norm-mult)
  also have exp (3 / 2 * pi * |Im s| - ln r * Re s) = exp (3 / 2 * pi * |Im s|
* r powr (-Re s))
    using r by (simp add: exp-diff powr-def exp-minus inverse-eq-divide)
  finally have norm (f s z) ≤ C * exp (3 / 2 * pi * |Im s|) * r powr (-Re s -
1) using r
    by (simp add: f-def exp-diff exp-minus field-simps powr-diff)
  also have ... ≤ max 0 (C * exp (3 / 2 * pi * |Im s|)) * r powr (-Re s - 1)
    by (intro mult-right-mono max.coboundedI2) auto
  finally have norm (f s z) ≤ ... .
}
with that[of max 0 (C * exp (3 / 2 * pi * |Im s|))] show ?thesis by auto
qed

```

We can now relate the integral along a partial Hankel contour that is cut off at  $-\pi$  to  $\zeta(1-s, a)/\Gamma(s)$ .

**lemma** *rGamma-hurwitz-zeta-eq-contour-integral*:

```

fixes s :: complex and r :: real
assumes s ≠ 0 and r: r ∈ {1..<2} and a: a > 0
defines err1 ≡ (λs r. contour-integral (part-circlepath 0 r pi 0) (f s))
defines err2 ≡ (λs r. cnj (contour-integral (part-circlepath 0 r pi 0) (f (cnj s))))
shows 2 * i * pi * rGamma s * hurwitz-zeta a (1 - s) =
      err2 s r - err1 s r + 2 * i * sin (pi * s) * (CLBINT x:{r..}. g s x)
(is ?f s = ?g s)
proof (rule analytic-continuation-open[where f = ?f])
  fix s :: complex assume s: s ∈ {s. Re s < 0}

```

— We first show that the integrals along the Hankel contour cut off at  $-\pi$  all have the same value, no matter what the radius of the circle is (as long as it is small enough). We call this value  $C$ .

This argument could be done by a homotopy argument, but it is easier to simply re-use the above result about the contour integral along the annulus where we fix the radius of the outer circle to  $\pi$ .

```

define C where C = -contour-integral (part-circlepath 0 pi 0 pi) (f s) +

```

```

      cnj (contour-integral (part-circlepath 0 pi 0 pi) (f (cnj s)))
have integrable: set-integrable lborel A (g s)
  if A ∈ sets lborel A ⊆ {0<..} for A
proof (rule set-integrable-subset)
  show set-integrable lborel {0<..} (g s)
    using Gamma-times-hurwitz-zeta-integrable[of 1 - s a] s a
    by (simp add: g-def exp-of-real exp-minus integrable-completion set-integrable-def)
qed (insert that, auto)

{
  fix r' :: real assume r': r' ∈ {0<.. $2$ }
  from hurwitz-formula-integral-semiannulus(2)[of r' s 0] and r'
  have f s contour-integrable-on part-circlepath 0 r' pi 0
    by (auto simp: hankel-semiannulus-def add-ac)
} note integrable-circle = this
{
  fix r' :: real assume r': r' ∈ {0<.. $2$ }
  from hurwitz-formula-integral-semiannulus(2)[of r' cnj s 0] and r'
  have f (cnj s) contour-integrable-on part-circlepath 0 r' pi 0
    by (auto simp: hankel-semiannulus-def add-ac)
} note integrable-circle' = this

have eq:  $-2 * i * \sin(\pi * s) * (CLBINT x:\{r..pi\}. g s x) + (err1 s r - err2 s r) = C$ 
  if r: r ∈ {0<.. $2$ } for r :: real
proof -
  have eq1:  $\int \{r..pi\} (\lambda x. cnj (x^{\text{powr}} - cnj s) * (\exp(-a * x))) / (1 - (\exp(-x))) = \int \{r..pi\} (g s)$  using r
    by (intro integral-cong) (auto simp: cnj-powr g-def exp-of-real exp-minus)
  have eq2:  $\int \{r..pi\} (\lambda x. cnj (x^{\text{powr}} - cnj s) * (\exp(a * x))) / (1 - (\exp x)) = \int \{r..pi\} (\lambda x. x^{\text{powr}} - s * (\exp(a * x))) / (1 - (\exp x))$  using
r
    by (intro integral-cong) (auto simp: cnj-powr)

from hurwitz-formula-integral-semiannulus(1)[of r s 0] hurwitz-formula-integral-semiannulus(1)[of
r cnj s 0]
  have  $\exp(-i * \pi * s) * \int \{r..real (2 * 0 + 1) * pi\} (g s) + \int \{r..real (2 * 0 + 1) * pi\} (\lambda x. x^{\text{powr}} - s * \exp(a * x)) / (1 - \exp x) +$ 
     $\text{contour-integral (part-circlepath 0 (real (2 * 0 + 1) * pi) 0 pi) (f s) +$ 
     $\text{contour-integral (part-circlepath 0 r pi 0) (f s) - cnj (}$ 
     $\exp(-i * \pi * cnj s) * \int \{r..real (2 * 0 + 1) * pi\} (\lambda x. x^{\text{powr}} - cnj s * \exp(-a * x)) / (1 - \exp(-x)) +$ 
     $\int \{r..real (2 * 0 + 1) * pi\} (\lambda x. x^{\text{powr}} - cnj s * \exp(a * x)) / (1 - \exp x) +$ 

```



$\text{contour-integral (part-circlepath 0 (real (2 * 0 + 1) * pi) 0 pi) (f (cnj s)) +$   
 $\text{contour-integral (part-circlepath 0 r pi 0) (f (cnj s))) = 0$  (**is** ?lhs = -)  
**unfolding g-def using r by** (subst (1 2) hurwitz-formula-integral-semiannulus)  
*auto*  
**also have** ?lhs =  $-2 * i * \sin(\pi * s) * \text{integral } \{r..pi\} (g s) + \text{err1 } s r - \text{err2 } s r - C$   
**using** eq1 eq2  
**by** (auto simp: integral-cnj exp-cnj err1-def err2-def sin-exp-eq algebra-simps C-def)  
**also have**  $\text{integral } \{r..pi\} (g s) = (\text{CLBINT } x:\{r..pi\}. g s x)$  **using** r  
**by** (intro set-borel-integral-eq-integral(2) [symmetric] integrable) *auto*  
**finally show**  $-2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{r..pi\}. g s x) + (\text{err1 } s r - \text{err2 } s r) = C$   
**by** (simp add: algebra-simps)  
**qed**

— Next, compute the value of  $C$  by letting the radius tend to 0 so that the contribution of the circle vanishes.

**have**  $((\lambda r. -2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{r..pi\}. g s x) + (\text{err1 } s r - \text{err2 } s r)) \longrightarrow$   
 $-2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{0<..pi\}. g s x) + 0)$  (*at-right 0*)  
**proof** (intro tendsto-intros tendsto-set-lebesgue-integral-at-right integrable)  
**from** hurwitz-formula-bound2[of s] **guess** C1 . **note** C1 = this  
**from** hurwitz-formula-bound2[of cnj s] **guess** C2 . **note** C2 = this  
**have** ev: eventually  $(\lambda r::\text{real}. r \in \{0 < .. < 2\})$  (*at-right 0*)  
**by** (intro eventually-at-right-real) *auto*  
**show**  $((\lambda r. \text{err1 } s r - \text{err2 } s r) \longrightarrow 0)$  (*at-right 0*)  
**proof** (rule Lim-null-comparison[OF eventually-mono[OF ev]])  
**fix** r :: real **assume** r:  $r \in \{0 < .. < 2\}$   
**have**  $\text{norm } (\text{err1 } s r - \text{err2 } s r) \leq \text{norm } (\text{err1 } s r) + \text{norm } (\text{err2 } s r)$   
**by** (rule norm-triangle-ineq4)  
**also have**  $\text{norm } (\text{err1 } s r) \leq C1 * r \text{ powr } (-\text{Re } s - 1) * r * |0 - pi|$   
**unfolding** err1-def **using** C1(1) C1(2)[of r] pi-gt3 integrable-circle[of r]  
path-image-part-circlepath-subset'[of r 0 pi 0] r  
**by** (intro contour-integral-bound-part-circlepath) *auto*  
**also have**  $\dots = C1 * r \text{ powr } (-\text{Re } s) * pi$  **using** r  
**by** (simp add: powr-diff field-simps)  
**also have**  $\text{norm } (\text{err2 } s r) \leq C2 * r \text{ powr } (-\text{Re } s - 1) * r * |0 - pi|$   
**unfolding** err2-def complex-mod-cnj **using** C2(1) C2(2)[of r] r  
pi-gt3 integrable-circle'[of r] path-image-part-circlepath-subset'[of r 0 pi 0]  
**by** (intro contour-integral-bound-part-circlepath) *auto*  
**also have**  $\dots = C2 * r \text{ powr } (-\text{Re } s) * pi$  **using** r  
**by** (simp add: powr-diff field-simps)  
**also have**  $C1 * r \text{ powr } (-\text{Re } s) * pi + C2 * r \text{ powr } (-\text{Re } s) * pi =$   
 $(C1 + C2) * pi * r \text{ powr } (-\text{Re } s)$  **by** (simp add: algebra-simps)  
**finally show**  $\text{norm } (\text{err1 } s r - \text{err2 } s r) \leq (C1 + C2) * pi * r \text{ powr } -\text{Re } s$   
**by** simp  
**next**

**show**  $((\lambda x. (C1 + C2) * \pi * x \text{ pow} - \text{Re } s) \longrightarrow 0)$  *(at-right 0)* **using**  $s$   
**by** *(auto intro!: tendsto-eq-intros simp: eventually-at exI[of - 1])*  
**qed**  
**qed auto**  
**moreover have** *eventually*  $(\lambda r::\text{real}. r \in \{0 < .. < 2\})$  *(at-right 0)*  
**by** *(intro eventually-at-right-real) auto*  
**hence eventually**  $(\lambda r. -2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{r.. \pi\}. g \ s \ x) +$   
 $(\text{err1 } s \ r - \text{err2 } s \ r) = C)$  *(at-right 0)* **by** *eventually-elim (use eq in auto)*  
**hence**  $((\lambda r. -2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{r.. \pi\}. g \ s \ x) + (\text{err1 } s \ r -$   
 $\text{err2 } s \ r)) \longrightarrow C)$   
*(at-right 0)* **by** *(rule tendsto-eventually)*  
**ultimately have**  $[\text{simp}]: C = -2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{0 < .. \pi\}. g \ s$   
 $x)$   
**using** *tendsto-unique by force*

— We now rearrange everything and obtain the result.

**have**  $2 * i * \sin(\pi * s) * ((\text{CLBINT } x:\{0 < .. \pi\}. g \ s \ x) - (\text{CLBINT } x:\{r.. \pi\}. g \ s \ x)) =$   
 $\text{err2 } s \ r - \text{err1 } s \ r$   
**using** *eq[of r] r by (simp add: algebra-simps)*  
**also have**  $\{0 < .. \pi\} = \{0 < .. < r\} \cup \{r.. \pi\}$  **using** *r pi-gt3 by auto*  
**also have**  $(\text{CLBINT } x:\{0 < .. < r\}. g \ s \ x) - (\text{CLBINT } x:\{r.. \pi\}. g \ s \ x) = (\text{CLBINT } x:\{0 < .. < r\}. g \ s \ x)$   
**using** *r pi-gt3 by (subst set-integral-Un[OF - integrable integrable]) auto*  
**also have**  $(\text{CLBINT } x:\{0 < .. < r\}. g \ s \ x) =$   
 $(\text{CLBINT } x:\{0 < .. < r\} \cup \{r.. \pi\}. g \ s \ x) - (\text{CLBINT } x:\{r.. \pi\}. g \ s \ x)$   
**using** *r pi-gt3 by (subst set-integral-Un[OF - integrable integrable]) auto*  
**also have**  $\{0 < .. < r\} \cup \{r.. \pi\} = \{0 < .. \pi\}$  **using** *r by auto*  
**also have**  $(\text{CLBINT } x:\{0 < .. \pi\}. g \ s \ x) = \text{Gamma}(1 - s) * \text{hurwitz-zeta } a(1 - s)$   
**using** *Gamma-times-hurwitz-zeta-integral[of 1 - s a] s a*  
**by** *(simp add: g-def exp-of-real exp-minus integral-completion set-lebesgue-integral-def)*  
**finally have**  $2 * i * (\sin(\pi * s) * \text{Gamma}(1 - s) * \text{hurwitz-zeta } a(1 - s) =$   
 $\text{err2 } s \ r - \text{err1 } s \ r + 2 * i * \sin(\pi * s) * (\text{CLBINT } x:\{r.. \pi\}. g \ s \ x))$   
**by** *(simp add: algebra-simps)*  
**also have**  $\sin(\pi * s) * \text{Gamma}(1 - s) = \pi * r\text{Gamma } s$   
**proof** *(cases s ∈ ℤ)*  
**case False**  
**with** *Gamma-reflection-complex[of s] show ?thesis*  
**by** *(auto simp: divide-simps sin-eq-0 Ints-def rGamma-inverse-Gamma mult-ac split: if-splits)*  
**next**  
**case True**  
**with**  $s$  **have**  $r\text{Gamma } s = 0$   
**by** *(auto simp: rGamma-eq-zero-iff nonpos-Ints-def Ints-def)*  
**moreover from**  $\text{True}$  **have**  $\sin(\pi * s) = 0$   
**by** *(subst sin-eq-0) (auto elim!: Ints-cases)*  
**ultimately show** *?thesis by simp*  
**qed**

**finally show**  $2 * i * pi * rGamma s * hurwitz-zeta a (1 - s) =$   
 $err2 s r - err1 s r + 2 * i * sin (pi * s) * (CLBINT x:\{r..\}. g s x)$   
**by** (*simp add: mult-ac*)

**next**  
 — By analytic continuation, we lift the result to the case of any non-zero  $s$ .  
**show**  $(\lambda s. 2 * i * pi * rGamma s * hurwitz-zeta a (1 - s))$  *holomorphic-on*  $-\{0\}$   
**using**  $a$   
**by** (*auto intro!: holomorphic-intros*)  
**show**  $(\lambda s. err2 s r - err1 s r + 2 * i * sin (pi * s) * (CLBINT x:\{r..\}. g s x))$   
*holomorphic-on*  $-\{0\}$   
**proof** (*intro holomorphic-intros*)  
**have**  $(\lambda s. err2 s r) = (\lambda s. - cnj (integral \{0..pi\} (h r (cnj s))))$  **using**  $r$   
**by** (*simp add: err2-def contour-integral-part-circlepath-reverse'*  
*contour-integral-part-circlepath-h*)  
**also have**  $(\lambda s. - cnj (integral \{0..pi\} (h r (cnj s)))) =$   
 $(\lambda s. (integral \{0..pi\} (\lambda x. h r s (-x))))$  **using**  $r$   
**by** (*simp add: integral-cnj h-def exp-cnj cis-cnj Ln-Reals-eq*)  
**also have**  $\dots = (\lambda s. integral \{-pi..0\} (h r s))$   
**by** (*subst Henstock-Kurzweil-Integration.integral-reflect-real [symmetric]*) *simp*  
**finally have**  $(\lambda s. err2 s r) = \dots$   
**moreover have**  $(\lambda s. integral \{-pi..0\} (h r s))$  *holomorphic-on*  $-\{0\}$   
**using**  $r$  **by** (*intro integral-h-holomorphic*) *auto*  
**ultimately show**  $(\lambda s. err2 s r)$  *holomorphic-on*  $-\{0\}$  **by** *simp*

**next**  
**have**  $(\lambda s. - integral \{0..pi\} (h r s))$  *holomorphic-on*  $-\{0\}$  **using**  $r$   
**by** (*intro holomorphic-intros integral-h-holomorphic*) *auto*  
**also have**  $(\lambda s. - integral \{0..pi\} (h r s)) = (\lambda s. err1 s r)$   
**unfolding** *err1-def* **using**  $r$   
**by** (*simp add: contour-integral-part-circlepath-reverse' contour-integral-part-circlepath-h*)  
**finally show**  $(\lambda s. err1 s r)$  *holomorphic-on*  $-\{0\}$  .

**next**  
**show**  $(\lambda s. CLBINT x:\{r..\}. g s x)$  *holomorphic-on*  $-\{0\}$   
**proof** (*rule holomorphic-on-balls-imp-entire'*)  
**fix**  $R :: real$   
**have** *eventually*  $(\lambda b. b > r)$  *at-top* **by** (*rule eventually-gt-at-top*)  
**hence**  $1: \text{eventually } (\lambda b. \text{continuous-on } (cball\ 0\ R)) (\lambda s. CLBINT x:\{r..b\}. g$   
 $s\ x) \wedge$   
 $(\lambda s. CLBINT x:\{r..b\}. g s x)$  *holomorphic-on ball 0 R*

*at-top*  
**proof** *eventually-elim*  
**case** (*elim b*)  
**have** *integrable: set-integrable lborel*  $\{r..b\}$   $(g s)$  **for**  $s$  **unfolding** *g-def* **using**  
 $r$   
**by** (*intro borel-integrable-atLeastAtMost' continuous-intros*) *auto*  
**have**  $(\lambda s. integral \{r..b\} (g s))$  *holomorphic-on UNIV* **using**  $r$   
**by** (*intro integral-g-holomorphic*) *auto*  
**also have**  $(\lambda s. integral \{r..b\} (g s)) = (\lambda s. CLBINT x:\{r..b\}. g s x)$   
**by** (*intro ext set-borel-integral-eq-integral(2)[symmetric] integrable*)  
**finally have**  $\dots$  *holomorphic-on UNIV* .

```

    thus ?case by (auto intro!: holomorphic-on-imp-continuous-on)
  qed

  have 2: uniform-limit (cball 0 R) ( $\lambda b s. \text{CLBINT } x:\{r..b\}. g s x$ )
    ( $\lambda s. \text{CLBINT } x:\{r..\}. g s x$ ) at-top
  proof (rule uniform-limit-set-lebesgue-integral-at-top)
    fix s :: complex and x :: real
    assume s:  $s \in \text{cball } 0 R$  and x:  $x \geq r$ 
    have norm (g s x) = x powr -Re s * exp (-a * x) / (1 - exp (-x)) using
x r
    by (simp add: g-def norm-mult norm-divide in-Reals-norm norm-powr-real-powr)
    also have ...  $\leq x \text{ powr } R * \exp (-a * x) / (1 - \exp (-x))$  using r s x
abs-Re-le-cmod[of s]
    by (intro mult-right-mono divide-right-mono powr-mono) auto
    finally show norm (g s x)  $\leq x \text{ powr } R * \exp (-a * x) / (1 - \exp (-x))$  .
  next
    show set-integrable lborel {r..} ( $\lambda x. x \text{ powr } R * \exp (-a * x) / (1 - \exp$ 
(-x)))
    using r a by (intro set-integrable-Gamma-hurwitz-aux2-real) auto
  qed (simp-all add: set-borel-measurable-def g-def)

  show ( $\lambda s. \text{CLBINT } x:\{r..\}. g s x$ ) holomorphic-on ball 0 R
    using holomorphic-uniform-limit[OF 1 2] by auto
  qed
  qed
  qed (insert ⟨s ≠ 0⟩,
    auto simp: connected-punctured-universe open-halfspace-Re-lt intro: exI[of -
-1])

  Finally, we obtain Hurwitz's formula by letting the radius of the outer circle
  tend to  $\infty$ .

  lemma hurwitz-zeta-formula-aux:
    fixes s :: complex
    assumes s:  $\text{Re } s > 1$ 
    shows  $r\text{Gamma } s * \text{hurwitz-zeta } a (1 - s) = (2 * \pi i) \text{ powr } -s *$ 
      ( $i \text{ powr } (-s) * F a s + i \text{ powr } s * F (-a) s$ )
  proof -
    from s have [simp]:  $s \neq 0$  by auto
    define r where r = (1 :: real)
    have r:  $r \in \{0 <..<2\}$  by (simp add: r-def)
    define R where R = ( $\lambda n. \text{real } (2 * n + 1) * \pi i$ )
    define bigc where bigc = ( $\lambda n. \text{contour-integral } (\text{part-circlepath } 0 (R n) 0 \pi i) (f$ 
s) -
       $\text{cnj } (\text{contour-integral } (\text{part-circlepath } 0 (R n) 0 \pi i) (f$ 
(cnj s))))
    define smallc where smallc =  $\text{contour-integral } (\text{part-circlepath } 0 r \pi i 0) (f s) -$ 
       $\text{cnj } (\text{contour-integral } (\text{part-circlepath } 0 r \pi i 0) (f (\text{cnj } s)))$ 
    define I where I = ( $\lambda n. \text{CLBINT } x:\{r..R n\}. g s x$ )

```

```

define F1 and F2 where
  F1 = ( $\lambda n. \exp(-s * \pi i * i / 2) * (\sum_{k \in \{0 <..n\}}. \exp(2 * \text{real } k * \pi i * a * i) * k \text{ powr } (-s))$ )
  F2 = ( $\lambda n. \exp(s * \pi i * i / 2) * (\sum_{k \in \{0 <..n\}}. \exp(2 * \text{real } k * \pi i * (-a) * i) * k \text{ powr } (-s))$ )

have R:  $R \geq \pi$  for n using r by (auto simp: R-def field-simps)
have [simp]:  $\neg(\pi \leq 0)$  using pi-gt-zero by linarith

have integrable: set-integrable lborel A (g s)
  if A  $\in$  sets lborel A  $\subseteq$   $\{r..\}$  for A
proof -
  have set-integrable lborel {r..} (g s)
    using set-integrable-Gamma-hurwitz-aux2[of r a s] a r
    by (simp add: g-def exp-of-real exp-minus)
  thus ?thesis by (rule set-integrable-subset) (use that in auto)
qed

{
  fix n :: nat
  from hurwitz-formula-integral-semiannulus(2)[of r s n] and r R[of n]
  have f s contour-integrable-on part-circlepath 0 (R n) 0 pi
    by (auto simp: hankel-semiannulus-def R-def add-ac)
} note integrable-circle = this
{
  fix n :: nat
  from hurwitz-formula-integral-semiannulus(2)[of r cnj s n] and r R[of n]
  have f (cnj s) contour-integrable-on part-circlepath 0 (R n) 0 pi
    by (auto simp: hankel-semiannulus-def R-def add-ac)
} note integrable-circle' = this

{
  fix n :: nat
  have ( $\exp(-i * \pi * s) * \text{integral } \{r..R \ n\} (g \ s) +$ 
     $\text{integral } \{r..R \ n\} (\lambda x. x \text{ powr } (-s) * \exp(a * x) / (1 - \exp x)) +$ 
     $\text{contour-integral } (\text{part-circlepath } 0 (R \ n) \ 0 \ \pi) (f \ s) +$ 
     $\text{contour-integral } (\text{part-circlepath } 0 \ r \ \pi \ 0) (f \ s) - \text{cnj} ($ 
     $\exp(-i * \pi * \text{cnj } \ s) * \text{integral } \{r..R \ n\} (g \ (\text{cnj } \ s)) +$ 
     $\text{integral } \{r..R \ n\} (\lambda x. x \text{ powr } (-\text{cnj } \ s) * \exp(a * x) / (1 - \exp x)) +$ 
     $\text{contour-integral } (\text{part-circlepath } 0 (R \ n) \ 0 \ \pi) (f \ (\text{cnj } \ s)) +$ 
     $\text{contour-integral } (\text{part-circlepath } 0 \ r \ \pi \ 0) (f \ (\text{cnj } \ s)))$ 
    =  $-2 * \pi * i * \exp(-s * \text{of-real } \pi * i / 2) * (\sum_{k \in \{0 <..n\}}. \text{Res } s \ k) -$ 
     $\text{cnj} (-2 * \pi * i * \exp(-\text{cnj } \ s * \text{of-real } \pi * i / 2) * (\sum_{k \in \{0 <..n\}}. \text{Res } s \ k))$ )
  (is ?lhs = ?rhs) unfolding R-def g-def using r
  by (subst (1 2) hurwitz-formula-integral-semiannulus) auto
  also have ?rhs =  $-2 * \pi * i * (\exp(-s * \pi * i / 2) * (\sum_{k \in \{0 <..n\}}. \text{Res } s \ k) +$ 

```

$$\exp (s * \pi * i / 2) * (\sum_{k \in \{0 <..n\}} \text{cnj} (\text{Res} (\text{cnj} s) k)))$$

**by** (*simp add: exp-cnj sum.distrib algebra-simps sum-distrib-left sum-distrib-right sum-negf*)

**also have**  $(\sum_{k \in \{0 <..n\}} \text{Res} s k) = (2 * \pi) \text{powr} (-s) * (\sum_{k \in \{0 <..n\}} \exp (2 * k * \pi * a * i) * k \text{powr} (-s))$

**(is - = ?S1)** **by** (*simp add: Res-def powr-times-real algebra-simps sum-distrib-left*)

**also have**  $(\sum_{k \in \{0 <..n\}} \text{cnj} (\text{Res} (\text{cnj} s) k)) = (2 * \pi) \text{powr} (-s) * (\sum_{k \in \{0 <..n\}} \exp (-2 * k * \pi * a * i) * k \text{powr} (-s))$

**by** (*simp add: Res-def cnj-powr powr-times-real algebra-simps exp-cnj sum-distrib-left*)

**also have**  $\exp (-s * \pi * i / 2) * ?S1 + \exp (s * \pi * i / 2) * \dots = (2 * \pi) \text{powr} (-s) * (\exp (-s * \pi * i / 2) * (\sum_{k \in \{0 <..n\}} \exp (2 * k * \pi * a * i) * k \text{powr} (-s)) + \exp (s * \pi * i / 2) * (\sum_{k \in \{0 <..n\}} \exp (-2 * k * \pi * a * i) * k \text{powr} (-s)))$

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

**also have 1:**  $\text{integral} \{r..R\} n \{g\} s = I n$  **unfolding** *I-def*

**by** (*intro set-borel-integral-eq-integral(2) [symmetric] integrable*) *auto*

**have 2:**  $\text{cnj} (\text{integral} \{r..R\} n \{g\} (\text{cnj} s)) = \text{integral} \{r..R\} n \{g\} s$  **using** *r* **unfolding** *integral-cnj* **by** (*intro integral-cong*) (*auto simp: g-def cnj-powr*)

**have 3:**  $\text{integral} \{r..R\} n \{ \lambda x. \exp (x * a) * \text{cnj} (x \text{powr} - \text{cnj} s) / (1 - \exp x) \} = \text{integral} \{r..R\} n \{ \lambda x. \exp (x * a) * \text{of-real } x \text{powr} - s / (1 - \exp x) \}$

**unfolding** *I-def g-def* **using** *r R[of n]* **by** (*intro integral-cong; force simp: cnj-powr*)*+*

**from 1 2 3** **have**  $?lhs = (\exp (-i * s * \pi) - \exp (i * s * \pi)) * I n + \text{bigc } n + \text{smallc}$

**by** (*simp add: integral-cnj cnj-powr algebra-simps exp-cnj bigc-def smallc-def g-def*)

**also have**  $\exp (-i * s * \pi) - \exp (i * s * \pi) = -2 * i * \sin (s * \pi)$

**by** (*simp add: sin-exp-eq' algebra-simps*)

**finally have**  $(-2 * i * \sin (s * \pi) * I n + \text{smallc}) + \text{bigc } n = -2 * i * \pi * (2 * \pi) \text{powr} (-s) * (F1 n + F2 n)$

**by** (*simp add: F1-F2-def algebra-simps*)

**} note** *eq = this*

**have**  $(\lambda n. -2 * i * \sin (s * \pi) * I n + \text{smallc} + \text{bigc } n) \longrightarrow (-2 * i * \sin (s * \pi)) * (\text{CLBINT } x:\{r..\}. g s x) + \text{smallc} + 0$

**unfolding** *I-def*

**proof** (*intro tendsto-intros filterlim-compose[OF tendsto-set-lebesgue-integral-at-top] integrable*)

**show** *filterlim R at-top sequentially* **unfolding** *R-def*

**by** (*intro filterlim-at-top-mult-tendsto-pos[OF tendsto-const] pi-gt-zero filterlim-compose[OF filterlim-real-sequentially] filterlim-subseq*) (*auto simp: strict-mono-Suc-iff*)

```

from hurwitz-formula-bound1[OF pi-gt-zero] guess C . note C = this
define D where D = C * exp (3 / 2 * pi * |Im s|)
from ⟨C ≥ 0⟩ have D ≥ 0 by (simp add: D-def)
show bigc ⟶ 0
proof (rule Lim-null-comparison[OF always-eventually[OF allI]])
  fix n :: nat
  have bound: norm (f s' z) ≤ D * R n powr (-Re s')
    if z: z ∈ sphere 0 (R n) Re s' = Re s |Im s'| = |Im s| for z s'
  proof -
    from z and r R[of n] have [simp]: z ≠ 0 by auto
    have not-in-ball: z ∉ ball (2 * m * pi * i) pi for m :: int
    proof -
      have dist z (2 * m * pi * i) ≥ |dist z 0 - dist 0 (2 * m * pi * i)|
        by (rule abs-dist-diff-le)
      also have dist 0 (2 * m * pi * i) = 2 * |m| * pi
        by (simp add: norm-mult)
      also from z have dist z 0 = R n by simp
      also have R n - 2 * |m| * pi = (int (2 * n + 1) - 2 * |m|) * pi
        by (simp add: R-def algebra-simps)
      also have |...| = |int (2 * n + 1) - 2 * |m|| * pi
        by (subst abs-mult) simp-all
      also have |int (2 * n + 1) - 2 * |m|| ≥ 1 by presburger
      hence ... * pi ≥ 1 * pi by (intro mult-right-mono) auto
      finally show ?thesis by (simp add: dist-commute)
    qed

  have norm (f s' z) = norm (exp (-Ln' z * s')) * norm (exp (a * z) / (1 -
exp z))
    by (simp add: f-def exp-diff norm-mult norm-divide mult-ac exp-minus
norm-inverse
      divide-simps del: norm-exp-eq-Re)
  also have ... ≤ norm (exp (-Ln' z * s')) * C using not-in-ball
    by (intro mult-left-mono C) auto
  also have norm (exp (-Ln' z * s')) =
    exp (Im s' * (Im (Ln (- (i * z))) + pi / 2)) / exp (Re s' * ln (R
n))
    using z r R[of n] pi-gt-zero
    by (simp add: Ln'-def norm-mult norm-divide exp-add exp-diff exp-minus
norm-inverse algebra-simps inverse-eq-divide)
  also have ... ≤ exp (3/2 * pi * |Im s'|) / exp (Re s' * ln (R n))
  proof (intro divide-right-mono, subst exp-le-cancel-iff)
    have Im s' * (Im (Ln (- (i * z))) + pi / 2) ≤ |Im s' * (Im (Ln (- (i *
z))) + pi / 2)|
      by (rule abs-ge-self)
    also have ... ≤ |Im s'| * (pi + pi / 2)
    unfolding abs-mult using mpi-less-Im-Ln[of - (i * z)] Im-Ln-le-pi[of
- (i * z)]
      by (intro mult-left-mono order.trans[OF abs-triangle-ineq] add-mono)

```

*auto*  
**finally show**  $\text{Im } s' * (\text{Im } (\text{Ln } (- (i * z))) + \text{pi} / 2) \leq 3/2 * \text{pi} * |\text{Im } s'|$   
**by** (*simp add: algebra-simps*)  
**qed** *auto*  
**also have**  $\text{exp } (\text{Re } s' * \text{ln } (R n)) = R n \text{ powr } \text{Re } s'$   
**using**  $r R[\text{of } n]$  **by** (*auto simp: powr-def*)  
**finally show**  $\text{norm } (f s' z) \leq D * R n \text{ powr } (-\text{Re } s')$  **using**  $\langle C \geq 0 \rangle$   
**by** (*simp add: that D-def powr-minus mult-right-mono mult-left-mono field-simps*)  
**qed**

**have**  $\text{norm } (\text{bigc } n) \leq \text{norm } (\text{contour-integral } (\text{part-circlepath } 0 (R n) 0 \text{ pi}) (f s)) +$   
 $\text{norm } (\text{cnj } (\text{contour-integral } (\text{part-circlepath } 0 (R n) 0 \text{ pi}) (f (\text{cnj } s))))$   
**(is - ≤ norm ?err1 + norm ?err2) unfolding bigc-def by** (*rule norm-triangle-ineq4*)  
**also have**  $\text{norm } ?err1 \leq D * R n \text{ powr } (-\text{Re } s) * R n * |\text{pi} - 0|$   
**using**  $\langle D \geq 0 \rangle$  **and**  $r R[\text{of } n]$  **and** *pi-gt3* **and** *integrable-circle* **and**  
 $\text{path-image-part-circlepath-subset}[\text{of } 0 \text{ pi } R n 0]$  **and**  $\text{bound}[\text{of } - s]$   
**by** (*intro contour-integral-bound-part-circlepath*) *auto*  
**also have**  $\dots = D * \text{pi} * R n \text{ powr } (1 - \text{Re } s)$  **using**  $r R[\text{of } n]$  *pi-gt3*  
**by** (*simp add: powr-diff field-simps powr-minus*)  
**also have**  $\text{norm } ?err2 \leq D * R n \text{ powr } (-\text{Re } s) * R n * |\text{pi} - 0|$   
**unfolding** *complex-mod-cnj*  
**using**  $\langle D \geq 0 \rangle$  **and**  $r R[\text{of } n]$  **and** *pi-gt3* **and** *integrable-circle'* $[\text{of } n]$  **and**  
 $\text{path-image-part-circlepath-subset}[\text{of } 0 \text{ pi } R n 0]$  **and**  $\text{bound}[\text{of } - \text{cnj } s]$   
**by** (*intro contour-integral-bound-part-circlepath*) *auto*  
**also have**  $\dots = D * \text{pi} * R n \text{ powr } (1 - \text{Re } s)$  **using**  $r R[\text{of } n]$  *pi-gt3*  
**by** (*simp add: powr-diff field-simps powr-minus*)  
**finally show**  $\text{norm } (\text{bigc } n) \leq 2 * D * \text{pi} * R n \text{ powr } (1 - \text{Re } s)$   
**by** *simp*

**next**  
**have** *filterlim*  $R$  *at-top at-top* **by** *fact*  
**hence**  $(\lambda x. 2 * D * \text{pi} * R x \text{ powr } (1 - \text{Re } s)) \longrightarrow 2 * D * \text{pi} * 0$  **using**  
*s unfolding R-def*  
**by** (*intro tendsto-intros tendsto-neg-powr*) *auto*  
**thus**  $(\lambda x. 2 * D * \text{pi} * R x \text{ powr } (1 - \text{Re } s)) \longrightarrow 0$  **by** *simp*  
**qed**

**qed** *auto*  
**also have**  $(\lambda n. -2 * i * \sin (s * \text{pi}) * I n + \text{smallc} + \text{bigc } n) =$   
 $(\lambda n. -2 * i * \text{pi} * (2 * \text{pi}) \text{ powr } -s * (F1 n + F2 n))$  **by** (*subst eq*)

*auto*  
**finally have**  $\dots \longrightarrow (-2 * i * \sin (s * \text{pi})) * (\text{CLBINT } x:\{r..\}. g s x) +$   
 $\text{smallc}$  **by** *simp*

**moreover have**  $(\lambda n. -2 * i * \text{pi} * (2 * \text{pi}) \text{ powr } -s * (F1 n + F2 n)) \longrightarrow$   
 $-2 * i * \text{pi} * (2 * \text{pi}) \text{ powr } -s *$   
 $(\text{exp } (-s * \text{pi} * i / 2) * F a s + \text{exp } (s * \text{pi} * i / 2) * F (-a) s)$   
**unfolding** *F1-F2-def F-def* **using**  $s$  **by** (*intro tendsto-intros sum-tendsto-fds-perzeta*)  
**ultimately have**  $-2 * i * \text{pi} * (2 * \text{pi}) \text{ powr } -s *$



$s) =$   
 $(\exp(-s * \pi i * i / 2) * F a s + \exp(s * \pi i * i / 2) * F(-a))$   
 $(-2 * i * \sin(s * \pi i)) * (CLBINT x:\{r..\}, g s x) + smallc$   
**by** (force intro: tendsto-unique)  
**also have**  $\dots = -2 * i * \pi * rGamma s * hurwitz-zeta a (1 - s)$  **using**  $s r a$   
**using**  $rGamma-hurwitz-zeta-eq-contour-integral[of s r]$   
**by** (simp add: r-def smallc-def algebra-simps)  
**also have**  $\exp(-s * \text{complex-of-real } \pi i * i / 2) = i \text{ powr } (-s)$   
**by** (simp add: powr-def field-simps)  
**also have**  $\exp(s * \text{complex-of-real } \pi i * i / 2) = i \text{ powr } s$   
**by** (simp add: powr-def field-simps)  
**finally show**  $rGamma s * hurwitz-zeta a (1 - s) = (2 * \pi i) \text{ powr } -s * (i \text{ powr } (-s) * F a s + i \text{ powr } s * F(-a) s)$  **by** simp  
**qed**  
**end**

We can now use Hurwitz's formula to prove the following nice formula that expresses the periodic zeta function in terms of the Hurwitz zeta function:

$$F(s, a) = (2\pi)^{s-1} i \Gamma(1-s) (i^{-s} \zeta(1-s, a) - i^s \zeta(1-s, 1-a))$$

This holds for all  $s$  with  $\mathit{mathfrak}\{R\}(s) > 0$  as long as  $a \notin \mathbb{Z}$ . For convenience, we move the  $\Gamma$  function to the left-hand side in order to avoid having to account for its poles.

**lemma** *perzeta-conv-hurwitz-zeta-aux*:

**fixes**  $a :: \text{real}$  **and**  $s :: \text{complex}$   
**assumes**  $a : a \in \{0 < .. < 1\}$  **and**  $s : \text{Re } s > 0$   
**shows**  $rGamma (1 - s) * \text{eval-fds } (fds-perzeta a) s = (2 * \pi i) \text{ powr } (s - 1) * i *$   
 $(i \text{ powr } -s * hurwitz-zeta a (1 - s) -$   
 $i \text{ powr } s * hurwitz-zeta (1 - a) (1 - s))$   
**(is ?lhs s = ?rhs s)**

**proof** (rule *analytic-continuation-open*[**where**  $f = ?lhs$ ])

**show** *connected*  $\{s. \text{Re } s > 0\}$   
**by** (intro *convex-connected convex-halfspace-Re-gt*)  
**show**  $\{s. \text{Re } s > 1\} \neq \{\}$  **by** (auto intro: *exI[of - 2]*)  
**show**  $(\lambda s. rGamma (1 - s) * \text{eval-fds } (fds-perzeta a) s)$  *holomorphic-on*  $\{s. 0 < \text{Re } s\}$

**unfolding** *perzeta-def* **using**  $a$   
**by** (auto intro!: *holomorphic-intros le-less-trans[OF conv-abscissa-perzeta]* elim!: *Ints-cases*)

**show** *?rhs holomorphic-on*  $\{s. 0 < \text{Re } s\}$  **using** *assms* **by** (auto intro!: *holomorphic-intros*)

**next**

**fix**  $s$  **assume**  $s : s \in \{s. \text{Re } s > 1\}$   
**have** [*simp*]:  $fds-perzeta (1 - a) = fds-perzeta (-a)$   
**using** *fds-perzeta.plus-of-nat[of -a 1]* **by** *simp*  
**have** [*simp*]:  $fds-perzeta (a - 1) = fds-perzeta a$

**using** *fds-perzeta.minus-of-nat[of a 1]* **by** *simp*  
**from** *s* **have** [*simp*]: *Gamma s ≠ 0* **by** (*auto simp: Gamma-eq-zero-iff elim!: nonpos-Ints-cases*)

**have**  $(2 * \pi) \text{ powr } (-s) * (i * (i \text{ powr } (-s) * (r\text{Gamma } s * \text{hurwitz-zeta } a (1 - s)) - i \text{ powr } s * (r\text{Gamma } s * \text{hurwitz-zeta } (1 - a) (1 - s)))) = (2 * \pi) \text{ powr } (-s) * ((i \text{ powr } (1 - s) * (r\text{Gamma } s * \text{hurwitz-zeta } a (1 - s))) + i \text{ powr } (s - 1) * (r\text{Gamma } s * \text{hurwitz-zeta } (1 - a) (1 - s)))$   
**by** (*simp add: powr-diff field-simps powr-minus*)  
**also have**  $\dots = ((2 * \pi) \text{ powr } (-s)) ^ 2 * (eval-fds (fds-perzeta a) s * (i \text{ powr } s * i \text{ powr } (s - 1) + i \text{ powr } (-s) * i \text{ powr } (1 - s)) + eval-fds (fds-perzeta (-a)) s * (i \text{ powr } s * i \text{ powr } (1 - s) + i \text{ powr } (-s) * i \text{ powr } (s - 1)))$   
**using** *s a* **by** (*subst (1 2) hurwitz-zeta-formula-aux*) (*auto simp: algebra-simps power2-eq-square*)  
**also have**  $(i \text{ powr } s * i \text{ powr } (1 - s) + i \text{ powr } (-s) * i \text{ powr } (s - 1)) = \exp(i * \text{complex-of-real } \pi / 2) + \exp(-i * \text{complex-of-real } \pi / 2)$   
**by** (*simp add: powr-def exp-add [symmetric] field-simps*)  
**also have**  $\dots = 0$  **by** (*simp add: exp-eq-polar*)  
**also have**  $i \text{ powr } s * i \text{ powr } (s - 1) = i \text{ powr } (2 * s - 1)$   
**by** (*simp add: powr-def exp-add [symmetric] field-simps*)  
**also have**  $i \text{ powr } (-s) * i \text{ powr } (1 - s) = i \text{ powr } (1 - 2 * s)$   
**by** (*simp add: powr-def exp-add [symmetric] field-simps*)  
**also have**  $i \text{ powr } (2 * s - 1) + i \text{ powr } (1 - 2 * s) = 2 * \cos((2 * s - 1) * \pi / 2)$   
**by** (*simp add: powr-def cos-exp-eq algebra-simps minus-divide-left cos-sin-eq*)  
**also have**  $\dots = 2 * \sin(\pi - s * \pi)$  **by** (*simp add: cos-sin-eq field-simps*)  
**also have**  $\dots = 2 * \sin(s * \pi)$  **by** (*simp add: sin-diff*)  
**finally have**  $i * (r\text{Gamma } s * i \text{ powr } (-s) * \text{hurwitz-zeta } a (1 - s) - r\text{Gamma } s * i \text{ powr } s * \text{hurwitz-zeta } (1 - a) (1 - s)) = 2 * (2 * \pi) \text{ powr } -s * \sin(s * \pi) * eval-fds (fds-perzeta a) s$   
**by** (*simp add: power2-eq-square mult-ac*)  
**hence**  $(2 * \pi) \text{ powr } s / 2 * i * (i \text{ powr } (-s) * \text{hurwitz-zeta } a (1 - s) - i \text{ powr } s * \text{hurwitz-zeta } (1 - a) (1 - s)) = \text{Gamma } s * \sin(s * \pi) * eval-fds (fds-perzeta a) s$   
**by** (*subst (asm) (2) powr-minus*) (*simp add: field-simps rGamma-inverse-Gamma*)  
**also have**  $\text{Gamma } s * \sin(s * \pi) = \pi * r\text{Gamma } (1 - s)$   
**using** *Gamma-reflection-complex[of s]*  
**by** (*auto simp: divide-simps rGamma-inverse-Gamma mult-ac split: if-splits*)  
**finally show** *?lhs s = ?rhs s* **by** (*simp add: powr-diff*)  
**qed** (*insert s, auto simp: open-halfspace-Re-gt*)

We can now use the above equation as a defining equation to continue the periodic zeta function  $F$  to the entire complex plane except at non-negative

integer values for  $s$ . However, the positive integers are already covered by the original Dirichlet series definition of  $F$ , so we only need to take care of  $s = 0$ . We do this by cancelling the pole of  $\Gamma$  at  $0$  with the zero of  $i^{-s}\zeta(1-s, a) - i^s\zeta(1-s, 1-a)$ .

**lemma**

**assumes**  $q' \notin \mathbb{Z}$   
**shows** *holomorphic-perzeta'*: *perzeta*  $q'$  *holomorphic-on*  $A$   
**and** *perzeta-altdef2*:  $\text{Re } s > 0 \implies \text{perzeta } q' s = \text{eval-fds } (\text{fds-perzeta } q') s$   
**proof** –  
**define**  $q$  **where**  $q = \text{frac } q'$   
**from** *assms* **have**  $q: q \in \{0 < \cdot < 1\}$  **by** (*auto simp: q-def frac-lt-1*)  
**hence** [*simp*]:  $q \notin \mathbb{Z}$  **by** (*auto elim!: Ints-cases*)  
**have** [*simp*]:  $\text{frac } q = q$  **by** (*simp add: q-def frac-def*)  
**define**  $f$  **where**  $f = (\lambda s. \text{complex-of-real } (2 * \pi i) \text{ powr } (s - 1) * i * \text{Gamma } (1 - s) * (i \text{ powr } (-s) * \text{hurwitz-zeta } q (1 - s) - i \text{ powr } s * \text{hurwitz-zeta } (1 - q) (1 - s)))$   
{  
**fix**  $s :: \text{complex}$  **assume**  $1 - s \in \mathbb{Z}_{\leq 0}$   
**then obtain**  $n$  **where**  $1 - s = \text{of-int } n$   $n \leq 0$  **by** (*auto elim!: nonpos-Ints-cases*)  
**hence**  $s = 1 - \text{of-int } n$  **by** (*simp add: algebra-simps*)  
**also have**  $\dots \in \mathbb{N}$  **using**  $\langle n \leq 0 \rangle$  **by** (*auto simp: Nats-altdef1 intro: exI[of - 1 - n]*)  
**finally have**  $s \in \mathbb{N}$  .  
} **note**  $*$  = *this*  
**hence**  $f$  *holomorphic-on*  $-\mathbb{N}$  **using**  $q$   
**by** (*auto simp: f-def Nats-altdef2 nonpos-Ints-altdef not-le intro!: holomorphic-intros*)  
**also have** *?this*  $\longleftrightarrow$  *perzeta*  $q$  *holomorphic-on*  $-\mathbb{N}$  **using** *assms*  
**by** (*intro holomorphic-cong refl*) (*auto simp: perzeta-def Let-def f-def*)  
**finally have** *holo*: *perzeta*  $q$  *holomorphic-on*  $-\mathbb{N}$  .  
**have** *f-altdef*:  $f s = \text{eval-fds } (\text{fds-perzeta } q) s$  **if**  $\text{Re } s > 0$  **and**  $s \notin \mathbb{N}$  **for**  $s$   
**using** *perzeta-conv-hurwitz-zeta-aux*[*OF*  $q$ , *of*  $s$ ] **that**  $*$   
**by** (*auto simp: rGamma-inverse-Gamma Gamma-eq-zero-iff divide-simps f-def perzeta-def split: if-splits*)  
**show** *perzeta*  $q' s = \text{eval-fds } (\text{fds-perzeta } q') s$  **if**  $\text{Re } s > 0$  **for**  $s$   
**using** *f-altdef*[*of*  $s$ ] **that** *assms* **by** (*auto simp: f-def perzeta-def Let-def q-def*)  
**have** *cont*: *isCont* (*perzeta*  $q$ )  $s$  **if**  $s \in \mathbb{N}$  **for**  $s$   
**proof** (*cases*  $s = 0$ )  
**case** *False*  
**with that** **obtain**  $n$  **where** [*simp*]:  $s = \text{of-nat } n$  **and**  $n: n > 0$   
**by** (*auto elim!: Nats-cases*)  
**have**  $*$ : *open* ( $\{s. \text{Re } s > 0\} - (\mathbb{N} - \{\text{of-nat } n\})$ ) **using** *Nats-subset-Ints*  
**by** (*intro open-Diff closed-subset-Ints open-halfspace-Re-gt*) *auto*  
**have** *eventually* ( $\lambda s. s \in \{s. \text{Re } s > 0\} - (\mathbb{N} - \{\text{of-nat } n\})$ ) (*nhds* (*of-nat*  $n$ ))

```

using ⟨ $n > 0$ ⟩
  by (intro eventually-nhds-in-open *) auto
  hence ev: eventually (λs. eval-fds (fds-perzeta q) s = perzeta q s) (nhds (of-nat
n))
  proof eventually-elim
    case (elim s)
    thus ?case using q f-altdef[of s]
      by (auto simp: perzeta-def dist-of-nat f-def elim!: Nats-cases Ints-cases)
  qed
  have isCont (eval-fds (fds-perzeta q)) (of-nat n) using q and ⟨ $n > 0$ ⟩
    by (intro continuous-eval-fds le-less-trans[OF conv-abscissa-perzeta'])
      (auto elim!: Ints-cases)
  also have ?this  $\longleftrightarrow$  isCont (perzeta q) (of-nat n) using ev
    by (intro isCont-cong ev)
  finally show ?thesis by simp
next
  assume [simp]: s = 0
  define a where a = Complex (ln q) (-pi / 2)
  define b where b = Complex (ln (1 - q)) (pi / 2)
  have eventually (λs::complex. s  $\notin$   $\mathbb{N}$ ) (at 0)
    unfolding eventually-at-topological using Nats-subset-Ints
    by (intro exI[of -  $\neg(\mathbb{N} - \{0\})$ ] conjI open-Compl closed-subset-Ints) auto
  hence ev: eventually (λs. perzeta q s = (2 * pi) powr (s - 1) * Gamma (1 -
s) * i *
      (i powr - s * pre-zeta q (1 - s) - i powr s * pre-zeta (1 - q) (1 -
s) +
      (exp (b * s) - exp (a * s)) / s)) (at (0::complex))
    (is eventually (λs. - = ?f s) -)
  proof eventually-elim
    case (elim s)
    have perzeta q s = (2 * pi) powr (s - 1) * Gamma (1 - s) * i *
      (i powr (-s) * hurwitz-zeta q (1 - s) -
      i powr s * hurwitz-zeta (1 - q) (1 - s)) (is - = - * ?T)
      using elim by (auto simp: perzeta-def powr-diff powr-minus field-simps)
    also have ?T = i powr (-s) * pre-zeta q (1 - s) - i powr s * pre-zeta (1 -
q) (1 - s) +
      (i powr s * (1 - q) powr s - i powr (-s) * q powr s) / s using
elim
      by (auto simp: hurwitz-zeta-def field-simps)
    also have i powr s * (1 - q) powr s = exp (b * s) using q
      by (simp add: powr-def exp-add algebra-simps Ln-Reals-eq Complex-eq b-def)
    also have i powr (-s) * q powr s = exp (a * s) using q
      by (simp add: powr-def exp-add Ln-Reals-eq exp-diff exp-minus diff-divide-distrib
ring-distrib inverse-eq-divide mult-ac Complex-eq a-def)
  finally show ?case .
qed
  have [simp]:  $\neg(pi \leq 0)$  using pi-gt-zero by (simp add: not-le)

```

**have**  $(\lambda s::\text{complex. if } s = 0 \text{ then } b - a \text{ else } (\exp (b * s) - \exp (a * s)) / s)$   
 $\text{has-fps-expansion } (\text{fps-exp } b - \text{fps-exp } a) / \text{fps-X}$  **(is ?f' has-fps-expansion**  
-)  
**by**  $(\text{rule fps-expansion-intros})+$   $(\text{auto intro!}:$   $\text{subdegree-geI simp: Ln-Reals-eq}$   
 $a\text{-def } b\text{-def})$   
**hence**  $\text{isCont } ?f' \ 0$  **by**  $(\text{rule has-fps-expansion-imp-continuous})$   
**hence**  $?f' - 0 \rightarrow b - a$  **by**  $(\text{simp add: isCont-def})$   
**also have**  $?this \longleftrightarrow (\lambda s. (\exp (b * s) - \exp (a * s)) / s) - 0 \rightarrow b - a$   
**by**  $(\text{intro filterlim-cong refl})$   $(\text{auto simp: eventually-at intro: exI[of - 1]})$   
**finally have**  $?f - 0 \rightarrow \text{of-real } (2 * \text{pi}) \text{ powr } (0 - 1) * \text{Gamma } (1 - 0) * i *$   
 $(i \text{ powr } -0 * \text{pre-zeta } q (1 - 0) - i \text{ powr } 0 * \text{pre-zeta } (1 - q) (1$   
 $- 0) + (b - a))$   
**(is filterlim - (nhds ?c) -)**  
**using**  $q$  **by**  $(\text{intro tendsto-intros isContD})$   
 $(\text{auto simp: complex-nonpos-Reals-iff intro!}:$   $\text{continuous-intros})$   
**also have**  $?c = \text{perzeta } q \ 0$  **using**  $q$   
**by**  $(\text{simp add: powr-minus perzeta-def Ln-Reals-eq a-def b-def}$   
 $\text{Complex-eq mult-ac inverse-eq-divide})$   
**also have**  $?f - 0 \rightarrow \dots \longleftrightarrow \text{perzeta } q - 0 \rightarrow \dots$   
**by**  $(\text{rule sym, intro filterlim-cong refl ev})$   
**finally show**  $\text{isCont } (\text{perzeta } q) \ s$  **by**  $(\text{simp add: isCont-def})$   
**qed**

**have**  $\text{perzeta } q \text{ field-differentiable at } s$  **for**  $s$   
**proof**  $(\text{cases } s \in \mathbb{N})$   
**case**  $\text{False}$   
**with**  $\text{holo have perzeta } q \text{ field-differentiable at } s \text{ within } -\mathbb{N}$   
**unfolding**  $\text{holomorphic-on-def}$  **by**  $\text{blast}$   
**also have**  $\text{at } s \text{ within } -\mathbb{N} = \text{at } s$  **using**  $\text{False}$   
**by**  $(\text{intro at-within-open})$   $\text{auto}$   
**finally show**  $?thesis .$   
**next**  
**case**  $\text{True}$   
**hence**  $*$ :  $\text{perzeta } q \text{ holomorphic-on } (\text{ball } s \ 1 - \{s\})$   
**by**  $(\text{intro holomorphic-on-subset[OF holo]})$   $(\text{auto elim!}:$   $\text{Nats-cases simp:}$   
 $\text{dist-of-nat})$   
**have**  $\text{perzeta } q \text{ holomorphic-on ball } s \ 1$  **using**  $\text{cont True}$   
**by**  $(\text{intro no-isolated-singularity'[OF - *]})$   
 $(\text{auto simp: at-within-open[of - ball } s \ 1] \text{ isCont-def})$   
**hence**  $\text{perzeta } q \text{ field-differentiable at } s \text{ within ball } s \ 1$   
**unfolding**  $\text{holomorphic-on-def}$  **by**  $\text{auto}$   
**thus**  $?thesis$  **by**  $(\text{simp add: at-within-open[of - ball } s \ 1])$   
**qed**  
**hence**  $\text{perzeta } q \text{ holomorphic-on UNIV}$   
**by**  $(\text{auto simp: holomorphic-on-def})$   
**also have**  $\text{perzeta } q = \text{perzeta } q'$  **by**  $(\text{simp add: q-def})$   
**finally show**  $\text{perzeta } q' \text{ holomorphic-on } A$  **by**  $\text{auto}$   
**qed**

**lemma** *perzeta-altdef1*:  $Re\ s > 1 \implies perzeta\ q'\ s = eval\ fds\ (fds\text{-}perzeta\ q')\ s$   
**by** (*cases*  $q' \in \mathbb{Z}$ ) (*auto simp: perzeta-int eval-fds-zeta fds-perzeta-int perzeta-altdef2*)

**lemma** *holomorphic-perzeta*:  $q \notin \mathbb{Z} \vee 1 \notin A \implies perzeta\ q\ holomorphic\text{-}on\ A$   
**by** (*cases*  $q \in \mathbb{Z}$ ) (*auto simp: perzeta-int intro: holomorphic-perzeta' holomorphic-zeta*)

**lemma** *holomorphic-perzeta''* [*holomorphic-intros*]:  
**assumes**  $f\ holomorphic\text{-}on\ A$  **and**  $q \notin \mathbb{Z} \vee (\forall x \in A. f\ x \neq 1)$   
**shows**  $(\lambda x. perzeta\ q\ (f\ x))\ holomorphic\text{-}on\ A$   
**proof** –  
**have**  $perzeta\ q \circ f\ holomorphic\text{-}on\ A$  **using** *assms*  
**by** (*intro holomorphic-on-compose holomorphic-perzeta*) *auto*  
**thus** *?thesis* **by** (*simp add: o-def*)  
**qed**

Using this analytic continuation of the periodic zeta function, Hurwitz's formula now holds (almost) on the entire complex plane.

**theorem** *hurwitz-zeta-formula*:  
**fixes**  $a :: real$  **and**  $s :: complex$   
**assumes**  $a \in \{0 < .. 1\}$  **and**  $s \neq 0$  **and**  $a \neq 1 \vee s \neq 1$   
**shows**  $rGamma\ s * hurwitz\text{-}zeta\ a\ (1 - s) =$   
 $(2 * pi)^{powr - s} * (i^{powr - s} * perzeta\ a\ s + i^{powr s} * perzeta\ (-a)$   
 $s)$   
**(is**  $?f\ s = ?g\ s)$   
**proof** –  
**define**  $A$  **where**  $A = UNIV - (if\ a \in \mathbb{Z}\ then\ \{0, 1\}\ else\ \{0 :: complex\})$   
**show** *?thesis*  
**proof** (*rule analytic-continuation-open[where f = ?f]*)  
**show**  $?f\ holomorphic\text{-}on\ A$  **using** *assms* **by** (*auto intro!: holomorphic-intros simp: A-def*)  
**show**  $?g\ holomorphic\text{-}on\ A$  **using** *assms*  
**by** (*auto intro!: holomorphic-intros simp: A-def minus-in-Ints-iff*)  
**next**  
**fix**  $s$  **assume**  $s \in \{s. Re\ s > 1\}$   
**thus**  $?f\ s = ?g\ s$  **using** *hurwitz-zeta-formula-aux[of a s] assms*  
**by** (*simp add: perzeta-altdef1*)  
**qed** (*insert assms, auto simp: open-halfspace-Re-gt A-def elim!: Ints-cases intro: connected-open-delete-finite exI[of - 2]*)  
**qed**

The equation expressing the periodic zeta function in terms of the Hurwitz zeta function can be extended similarly.

**theorem** *perzeta-conv-hurwitz-zeta*:  
**fixes**  $a :: real$  **and**  $s :: complex$   
**assumes**  $a \in \{0 < .. < 1\}$  **and**  $s \neq 0$   
**shows**  $rGamma\ (1 - s) * perzeta\ a\ s =$   
 $(2 * pi)^{powr (s - 1)} * i * (i^{powr (-s)} * hurwitz\text{-}zeta\ a\ (1 - s) -$   
 $i^{powr s} * hurwitz\text{-}zeta\ (1 - a)\ (1 - s))$

```

(is ?f s = ?g s)
proof (rule analytic-continuation-open[where f = ?f])
  show ?f holomorphic-on -{0} using assms by (auto intro!: holomorphic-intros
elim: Ints-cases)
  show ?g holomorphic-on -{0} using assms by (auto intro!: holomorphic-intros)
next
  fix s assume s ∈ {s. Re s > 1}
  thus ?f s = ?g s using perzeta-conv-hurwitz-zeta-aux[of a s] assms
  by (simp add: perzeta-altdef1)
qed (insert assms, auto simp: open-halfspace-Re-gt connected-punctured-universe
intro: exI[of - 2])

```

As a simple corollary, we derive the reflection formula for the Riemann zeta function:

```

corollary zeta-reflect:
  fixes s :: complex
  assumes s ≠ 0 s ≠ 1
  shows rGamma s * zeta (1 - s) = 2 * (2 * pi) powr -s * cos (s * pi / 2) *
zeta s
  using hurwitz-zeta-formula[of 1 s] assms
  by (simp add: zeta-def cos-exp-eq powr-def perzeta-int algebra-simps)

```

```

corollary zeta-reflect':
  fixes s :: complex
  assumes s ≠ 0 s ≠ 1
  shows rGamma (1 - s) * zeta s = 2 * (2 * pi) powr (s - 1) * sin (s * pi /
2) * zeta (1 - s)
  using zeta-reflect[of 1 - s] assms by (simp add: cos-sin-eq field-simps)

```

It is now easy to see that all the non-trivial zeroes of the Riemann zeta function must lie the critical strip  $(0; 1)$ , and they must be symmetric around the  $\Re(z) = \frac{1}{2}$  line.

```

corollary zeta-zeroD:
  assumes zeta s = 0 s ≠ 1
  shows Re s ∈ {0 <..<1} ∨ (∃ n::nat. n > 0 ∧ even n ∧ s = -real n)
proof (cases Re s ≤ 0)
  case False
  with zeta-Re-ge-1-nonzero[of s] assms have Re s < 1
  by (cases Re s < 1) auto
  with False show ?thesis by simp
next
  case True
  {
  assume *: ∧n. n > 0 ⇒ even n ⇒ s ≠ -real n
  have s ≠ of-int n for n :: int
  proof
  assume [simp]: s = of-int n
  show False
  proof (cases n 0::int rule: linorder-cases)

```

```

assume  $n < 0$ 
show False
proof (cases even n)
  case True
    hence  $\text{nat } (-n) > 0 \text{ even } (\text{nat } (-n))$  using  $\langle n < 0 \rangle$ 
    by (auto simp: even-nat-iff)
    with * have  $s \neq -\text{real } (\text{nat } (-n))$  .
    with  $\langle n < 0 \rangle$  and True show False by auto
  next
    case False
    with  $\langle n < 0 \rangle$  have  $\text{of-int } n = (-\text{of-nat } (\text{nat } (-n)) :: \text{complex})$  by simp
    also have  $\text{zeta } \dots = -(\text{bernoulli}' (\text{Suc } (\text{nat } (-n)))) / \text{of-nat } (\text{Suc } (\text{nat } (-n)))$ 
    using  $\langle n < 0 \rangle$  by (subst zeta-neg-of-nat) (auto)
    finally have  $\text{bernoulli}' (\text{Suc } (\text{nat } (-n))) = 0$  using assms
    by (auto simp del: of-nat-Suc)
    with False and  $\langle n < 0 \rangle$  show False
    by (auto simp: bernoulli'-zero-iff even-nat-iff)
  qed
qed (insert assms True, auto)
qed
hence  $r\text{Gamma } s \neq 0$ 
  by (auto simp: rGamma-eq-zero-iff nonpos-Ints-def)
moreover from assms have [simp]:  $s \neq 0$  by auto
ultimately have  $\text{zeta } (1 - s) = 0$  using zeta-reflect[of s] and assms
  by auto
with True zeta-Re-ge-1-nonzero[of 1 - s] have  $\text{Re } s > 0$  by auto
}
with True show ?thesis by auto
qed

```

**lemma** *zeta-zero-reflect*:

```

assumes  $\text{Re } s \in \{0 < .. < 1\}$  and  $\text{zeta } s = 0$ 
shows  $\text{zeta } (1 - s) = 0$ 

```

**proof** –

```

from assms have  $r\text{Gamma } s \neq 0$ 
  by (auto simp: rGamma-eq-zero-iff elim!: nonpos-Ints-cases)
moreover from assms have  $s \neq 0$  and  $s \neq 1$  by auto
ultimately show ?thesis using zeta-reflect[of s] and assms by auto

```

**qed**

**corollary** *zeta-zero-reflect-iff*:

```

assumes  $\text{Re } s \in \{0 < .. < 1\}$ 
shows  $\text{zeta } (1 - s) = 0 \iff \text{zeta } s = 0$ 
using zeta-zero-reflect[of s] zeta-zero-reflect[of 1 - s] assms by auto

```

## 2.10 More functional equations

**lemma** *perzeta-conv-hurwitz-zeta-multiplication*:



```

fixes  $k :: \text{nat}$  and  $a :: \text{int}$  and  $s :: \text{complex}$ 
assumes  $k > 0$   $s \neq 1$ 
shows  $k \text{ powr } s * \text{perzeta } (a / k) s =$ 
 $(\sum n=1..k. \exp (2 * \text{pi} * n * a / k * i) * \text{hurwitz-zeta } (n / k) s)$ 
(is ?lhs s = ?rhs s)
proof (rule analytic-continuation-open[where  $?f = ?lhs$  and  $?g = ?rhs$ ])
show connected ( $\{-1 :: \text{complex}\}$ ) by (rule connected-punctured-universe) auto
show  $\{s. \text{Re } s > 1\} \neq \{\}$  by (auto intro!: exI[of - 2])
next
fix  $s$  assume  $s: s \in \{s. \text{Re } s > 1\}$ 
let  $?f = \lambda n. \exp (2 * \text{pi} * n * a / k * i)$ 

show  $?lhs s = ?rhs s$ 
proof (rule sums-unique2)
have  $(\lambda m. \sum n=1..k. ?f n * (\text{of-nat } m + \text{of-real } (\text{real } n / \text{real } k)) \text{ powr } -s)$ 
sums
 $(\sum n=1..k. ?f n * \text{hurwitz-zeta } (\text{real } n / \text{real } k) s)$ 
using assms s by (intro sums-sum sums-mult sums-hurwitz-zeta) auto
also have  $(\lambda m. \sum n=1..k. ?f n * (\text{of-nat } m + \text{of-real } (\text{real } n / \text{real } k)) \text{ powr } -s) =$ 
 $(\lambda m. \text{of-nat } k \text{ powr } s * (\sum n=1..k. ?f n * \text{of-nat } (m * k + n) \text{ powr } -s))$ 
unfolding sum-distrib-left
proof (intro ext sum.cong, goal-cases)
case  $(2 m n)$ 
hence  $m * k + n > 0$  by (intro add-nonneg-pos) auto
hence  $\text{of-nat } 0 \neq (\text{of-nat } (m * k + n) :: \text{complex})$  by (simp only: of-nat-eq-iff)
also have  $\text{of-nat } (m * k + n) = \text{of-nat } m * \text{of-nat } k + (\text{of-nat } n :: \text{complex})$ 
by simp
finally have  $\text{nz}: \dots \neq 0$  by auto

have  $\text{of-nat } m + \text{of-real } (\text{real } n / \text{real } k) =$ 
 $(\text{inverse } (\text{of-nat } k) * \text{of-nat } (m * k + n) :: \text{complex})$  using assms

by (simp add: field-simps del: div-mult-self1 div-mult-self2 div-mult-self3
div-mult-self4)
also from nz have  $\dots \text{ powr } -s = \text{of-nat } k \text{ powr } s * \text{of-nat } (m * k + n) \text{ powr } -s$ 
by (subst powr-times-real) (auto simp: add-eq-0-iff powr-def exp-minus
Ln-inverse)
finally show  $?case$  by simp
qed auto
finally show  $\dots \text{ sums } (\sum n=1..k. ?f n * \text{hurwitz-zeta } (\text{real } n / \text{real } k) s) .$ 
next
define  $g$  where  $g = (\lambda m. \exp (2 * \text{pi} * i * m * (\text{real-of-int } a / \text{real } k)))$ 
have  $(\lambda m. g (Suc m) / (Suc m) \text{ powr } s) \text{ sums eval-fds } (\text{fds-perzeta } (a / k)) s$ 
unfolding g-def using  $s$  by (intro sums-fds-perzeta) auto
also have  $(\lambda m. g (Suc m) / (Suc m) \text{ powr } s) = (\lambda m. ?f (Suc m) * (Suc m) \text{ powr } -s)$ 

```

by (simp add: powr-minus field-simps g-def)  
 also have eval-fds (fds-perzeta (a / k)) s = perzeta (a / k) s  
 using s by (simp add: perzeta-altdef1)  
 finally have  $(\lambda m. \sum_{n=m*k..<m*k+k.} ?f (Suc n) * of-nat (Suc n) powr -s)$   
*sums perzeta (a / k) s*  
 using  $\langle k > 0 \rangle$  by (rule sums-group)  
 also have  $(\lambda m. \sum_{n=m*k..<m*k+k.} ?f (Suc n) * of-nat (Suc n) powr -s) =$   
 $(\lambda m. \sum_{n=1..k.} ?f (m * k + n) * of-nat (m * k + n) powr -s)$   
 proof (rule ext, goal-cases)  
 case (1 m)  
 show ?case using assms  
 by (intro ext sum.reindex-bij-witness[of -  $\lambda n. m * k + n - 1$   $\lambda n. Suc n -$   
 $m * k$ ]) auto  
 qed  
 also have  $(\lambda m n. ?f (m * k + n)) = (\lambda m n. ?f n)$   
 proof (intro ext)  
 fix m n :: nat  
 have  $?f (m * k + n) / ?f n = exp (2 * pi * m * a * i)$   
 using  $\langle k > 0 \rangle$  by (auto simp: ring-distrib add-divide-distrib exp-add mult-ac)  
 also have  $\dots = cis (2 * pi * (m * a))$   
 by (simp add: exp-eq-polar mult-ac)  
 also have  $\dots = 1$   
 by (rule cis-multiple-2pi) auto  
 finally show  $?f (m * k + n) = ?f n$   
 by simp  
 qed  
 finally show  $(\lambda m. of-nat k powr s * (\sum_{n=1..k.} ?f n * of-nat (m * k + n)$   
*powr -s)) sums*  
 $(of-nat k powr s * perzeta (a / k) s)$  by (rule sums-mult)  
 qed  
 qed (use assms in  $\langle$ auto intro!: holomorphic-intros simp: finite-imp-closed open-halfspace-Re-gt $\rangle$ )

**lemma perzeta-conv-hurwitz-zeta-multiplication<sup>1</sup>:**  
 fixes k :: nat and a :: int and s :: complex  
 assumes  $k > 0$   $s \neq 1$   
 shows  $perzeta (a / k) s = k powr -s *$   
 $(\sum_{n=1..k.} exp (2 * pi * n * a / k * i) * hurwitz-zeta (n / k) s)$   
 using perzeta-conv-hurwitz-zeta-multiplication[of k s a] assms  
 by (simp add: powr-minus field-simps)

**lemma zeta-conv-hurwitz-zeta-multiplication:**  
 fixes k a :: nat and s :: complex  
 assumes  $k > 0$   $s \neq 1$   
 shows  $k powr s * zeta s = (\sum_{n=1..k.} hurwitz-zeta (n / k) s)$   
 using perzeta-conv-hurwitz-zeta-multiplication[of k s 0]  
 using assms by (simp add: perzeta-int)

**lemma hurwitz-zeta-one-half-left:**  
 assumes  $s \neq 1$

**shows**  $\text{hurwitz-zeta } (1 / 2) s = (2 \text{ powr } s - 1) * \text{zeta } s$   
**using**  $\text{zeta-conv-hurwitz-zeta-multiplication[of } 2 s]$   $\text{assms}$   
**by** ( $\text{simp add: eval-nat-numeral zeta-def field-simps}$ )

**theorem**  $\text{hurwitz-zeta-functional-equation:}$

**fixes**  $h k :: \text{nat}$  **and**  $s :: \text{complex}$

**assumes**  $hk: k > 0$   $h \in \{0 <..k\}$  **and**  $s: s \notin \{0, 1\}$

**defines**  $a \equiv \text{real } h / \text{real } k$

**shows**  $r\text{Gamma } s * \text{hurwitz-zeta } a (1 - s) =$

$2 * (2 * \text{pi} * k) \text{ powr } -s *$

$(\sum n=1..k. \cos (s*\text{pi}/2 - 2*\text{pi}*n*h/k) * \text{hurwitz-zeta } (n / k) s)$

**proof** –

**from**  $hk$  **have**  $a: a \in \{0 <..1\}$  **by** ( $\text{auto simp: a-def}$ )

**have**  $r\text{Gamma } s * \text{hurwitz-zeta } a (1 - s) =$

$(2 * \text{pi}) \text{ powr } -s * (\text{i powr } -s * \text{perzeta } a s + \text{i powr } s * \text{perzeta } (-a) s)$

**using**  $s a$  **by** ( $\text{intro hurwitz-zeta-formula}$ )  $\text{auto}$

**also have**  $\dots = (2 * \text{pi}) \text{ powr } -s * (\text{i powr } -s * \text{perzeta } (\text{of-int } (\text{int } h) / k) s$

+

$\text{i powr } s * \text{perzeta } (\text{of-int } (-\text{int } h) / k) s)$

**by** ( $\text{simp add: a-def}$ )

**also have**  $\dots = (2 * \text{pi}) \text{ powr } -s * k \text{ powr } -s *$

$(\sum n=1..k. \text{i powr } -s * \text{cis } (2 * \text{pi} * n * h / k) * \text{hurwitz-zeta } (n / k) s) +$

$(\sum n=1..k. \text{i powr } s * \text{cis } (-2 * \text{pi} * n * h / k) * \text{hurwitz-zeta } (n / k) s)$

( $\text{is - = - * (?S1 + ?S2)}$ ) **using**  $hk a s$

**by** ( $\text{subst } (1 2) \text{perzeta-conv-hurwitz-zeta-multiplication}^{\wedge}$ )

( $\text{auto simp: field-simps sum-distrib-left sum-distrib-right exp-eq-polar}$ )

**also have**  $(2 * \text{pi}) \text{ powr } -s * k \text{ powr } -s = (2 * k * \text{pi}) \text{ powr } -s$

**using**  $hk \text{pi-gt-zero}$

**by** ( $\text{simp add: powr-def Ln-times-Reals field-simps exp-add exp-diff exp-minus}$ )

**also have**  $?S1 + ?S2 = (\sum n=1..k. (\text{i powr } -s * \text{cis } (2*\text{pi}*n*h/k) + \text{i powr } s$   
 $* \text{cis } (-2*\text{pi}*n*h/k)) *$

$\text{hurwitz-zeta } (n / k) s)$

( $\text{is - = } (\sum n \in -. ?c n * -)$ ) **by** ( $\text{simp add: algebra-simps sum.distrib}$ )

**also have**  $?c = (\lambda n. 2 * \cos (s*\text{pi}/2 - 2*\text{pi}*n*h/k))$

**proof**

**fix**  $n :: \text{nat}$

**have**  $\text{i powr } -s * \text{cis } (2*\text{pi}*n*h/k) = \text{exp } (-s*\text{pi}/2*\text{i} + 2*\text{pi}*n*h/k*\text{i})$

**unfolding**  $\text{exp-add}$  **by** ( $\text{simp add: powr-def cis-conv-exp mult-ac}$ )

**moreover have**  $\text{i powr } s * \text{cis } (-2*\text{pi}*n*h/k) = \text{exp } (s*\text{pi}/2*\text{i} + -2*\text{pi}*n*h/k*\text{i})$

**unfolding**  $\text{exp-add}$  **by** ( $\text{simp add: powr-def cis-conv-exp mult-ac}$ )

**ultimately have**  $?c n = \text{exp } (\text{i} * (s*\text{pi}/2 - 2*\text{pi}*n*h/k)) + \text{exp } (-\text{i} * (s*\text{pi}/2$   
 $- 2*\text{pi}*n*h/k))$

**by** ( $\text{simp add: mult-ac ring-distrib}$ )

**also have**  $\dots / 2 = \cos (s*\text{pi}/2 - 2*\text{pi}*n*h/k)$

**by** ( $\text{rule cos-exp-eq [symmetric]}$ )

**finally show**  $?c n = 2 * \cos (s*\text{pi}/2 - 2*\text{pi}*n*h/k)$

**by**  $\text{simp}$

**qed**

also have  $(2 * k * pi) \text{ powr } -s * (\sum_{n=1..k} \dots n * \text{hurwitz-zeta } (n / k) s) =$   
 $2 * (2 * pi * k) \text{ powr } -s * (\sum_{n=1..k} \cos (s*pi/2 - 2*pi*n*h/k) * \text{hurwitz-zeta } (n / k) s)$   
 by (simp add: sum-distrib-left sum-distrib-right mult-ac)  
 finally show ?thesis .  
 qed

lemma perzeta-one-half-left:  $s \neq 1 \implies \text{perzeta } (1 / 2) s = (2 \text{ powr } (1-s) - 1) * \text{zeta } s$   
 using perzeta-conv-hurwitz-zeta-multiplication'[of 2 s 1]  
 by (simp add: eval-nat-numeral hurwitz-zeta-one-half-left powr-minus field-simps zeta-def powr-diff)

lemma perzeta-one-half-left':  
 $\text{perzeta } (1 / 2) s =$   
 $(\text{if } s = 1 \text{ then } -\ln 2 \text{ else } (2 \text{ powr } (1 - s) - 1) / (s - 1)) * ((s - 1) * \text{pre-zeta } 1 s + 1)$   
 by (cases s = 1) (auto simp: perzeta-one-half-left field-simps zeta-def hurwitz-zeta-def)  
 end

### 3 The Laurent series expansion of $\zeta$ at 1

theory Zeta-Laurent-Expansion  
 imports Zeta-Function  
 begin

In this section, we shall derive the Laurent series expansion of  $\zeta(s)$  at  $s = 1$ , which is of the form

$$\zeta(s) = \frac{1}{s-1} + \sum_{n=0}^{\infty} \frac{(-1)^n \gamma_n}{n!} (s-1)^n$$

where the  $\gamma_n$  are the *Stieltjes constants*. Notably,  $\gamma_0$  is equal to the Euler–Mascheroni constant  $\gamma$ .

#### 3.1 Definition of the Stieltjes constants

We define the Stieltjes constants by their infinite series form, since it is fairly easy to show the convergence of the series by the comparison test.

definition stieltjes-gamma :: nat  $\Rightarrow$  'a :: real-algebra-1 where  
 stieltjes-gamma n =  
 $\text{of-real } (\sum_{k=1}^n k \cdot \ln (k+1) \wedge n / (k+1) - (\ln (k+2) \wedge (n+1) - \ln (k+1) \wedge (n+1))) / (n+1)$

lemma stieltjes-gamma-0 [simp]:  $\text{stieltjes-gamma } 0 = \text{euler-mascheroni}$   
 using euler-mascheroni-sum-real by (simp add: sums-iff stieltjes-gamma-def field-simps)

**lemma** *stieltjes-gamma-summable*:

$\text{summable } (\lambda k. \ln (k+1) \wedge n / (k+1) - (\ln (k+2) \wedge (n+1) - \ln (k+1) \wedge (n+1)) / (n+1))$   
*(is summable ?f)*

**proof** (*rule summable-comparison-test-bigo*)

**have** *eventually*  $(\lambda x::\text{real}. \ln x \wedge n - \ln x \wedge (n+1) * (\text{inverse } (\ln x) * (1 + \text{real } n))) *$

*inverse (real n + 1) = 0* *at-top*

**using** *eventually-gt-at-top*[*of 1*] **by** *eventually-elim* (*auto simp: field-simps*)

**thus**  $?f \in O(\lambda k. k \text{ powr } (-3/2))$

**by** *real-asymp*

**qed** (*simp-all add: summable-real-powr-iff*)

**lemma** *of-real-stieltjes-gamma* [*simp*]: *of-real (stieltjes-gamma k) = stieltjes-gamma k*

**by** (*simp add: stieltjes-gamma-def*)

**lemma** *sums-stieltjes-gamma*:

$(\lambda k. \ln (k+1) \wedge n / (k+1) - (\ln (k+2) \wedge (n+1) - \ln (k+1) \wedge (n+1)) / (n+1))$

*sums stieltjes-gamma n*

**using** *stieltjes-gamma-summable*[*of n*] **unfolding** *stieltjes-gamma-def* **by** (*simp add: summable-sums*)

We can now derive the alternative definition of the Stieltjes constants as a limit. This limit can also be written in the Euler–MacLaurin-style form

$$\lim_{m \rightarrow \infty} \left( \sum_{k=1}^m \frac{\ln^n k}{k} - \int_1^m \frac{\ln^n x}{x} dx \right),$$

which is perhaps a bit more illuminating.

**lemma** *stieltjes-gamma-real-limit-form*:

$(\lambda m. (\sum k = 1..m. \ln (\text{real } k) \wedge n / \text{real } k) - \ln (\text{real } m) \wedge (n+1) / \text{real } (n+1))$

$\longrightarrow$  *stieltjes-gamma n*

**proof** –

**have**  $(\lambda m::\text{nat}. \sum k < m. \ln (k+1) \wedge n / (k+1) - (\ln (k+2) \wedge (n+1) - \ln (k+1) \wedge (n+1)) / (n+1))$

$\longrightarrow$  *stieltjes-gamma n*

**using** *sums-stieltjes-gamma*[*of n*] **by** (*simp add: add-ac sums-def*)

**also have**  $(\lambda m::\text{nat}. \sum k < m. \ln (k+1) \wedge n / (k+1) - (\ln (k+2) \wedge (n+1) - \ln (k+1) \wedge (n+1)) / (n+1)) =$

$(\lambda m::\text{nat}. (\sum k=1..m. \ln k \wedge n / k) - \ln (m+1) \wedge (n+1) / (n+1))$

*(is ?lhs = ?rhs)*

**proof** (*rule ext, goal-cases*)

**fix**  $m :: \text{nat}$

**have**  $(\sum k < m. \ln (k+1) \wedge n / (k+1) - (\ln (k+2) \wedge (n+1) - \ln (k+1) \wedge (n+1))) / (n+1) =$   
 $(\sum k < m. \ln (k+1) \wedge n / (k+1)) -$   
 $(\sum k < m. \ln (Suc\ k+1) \wedge (n+1) - \ln (k+1) \wedge (n+1)) / (n+1)$   
**by** (*simp add: sum-subtractf flip: sum-divide-distrib*)  
**also have**  $(\sum k < m. \ln (k+1) \wedge n / (k+1)) = (\sum k = 1..m. \ln k \wedge n / k)$   
**by** (*rule sum.reindex-bij-witness[of -  $\lambda k. k-1$  Suc] auto*)  
**also have**  $(\sum k < m. \ln (Suc\ k+1) \wedge (n+1) - \ln (k+1) \wedge (n+1)) = \ln (m+1) \wedge (n+1)$   
**by** (*subst sum-lessThan-telescope*) *simp-all*  
**finally show** *?lhs m = ?rhs m .*  
**qed**  
**finally have**  $*$ :  $(\lambda m. (\sum k = 1..m. \ln k \wedge n / k) - \ln (m+1) \wedge (n+1) / (n+1))$   
 $\longrightarrow$  *stieltjes-gamma n .*  
**have**  $**$ :  $(\lambda m. \ln (m+1) \wedge (n+1) / (n+1) - \ln m \wedge (n+1) / (n+1))$   
 $\longrightarrow 0$   
**by** *real-asymp*  
**from** *tendsto-add[OF \* \*\*]* **show** *?thesis by (simp add: algebra-simps)*  
**qed**

**lemma** *stieltjes-gamma-limit-form*:

$(\lambda m. \text{of-real } ((\sum k = 1..m. \ln (\text{real } k) \wedge n / \text{real } k) - \ln (\text{real } m) \wedge (n+1) / \text{real } (n+1)))$   
 $\longrightarrow (\text{stieltjes-gamma } n :: 'a :: \text{real-normed-algebra-1})$

**proof** –

**have**  $(\lambda m. \text{of-real } ((\sum k = 1..m. \ln (\text{real } k) \wedge n / \text{real } k) - \ln m \wedge (n+1) / \text{real } (n+1)))$   
 $\longrightarrow (\text{of-real } (\text{stieltjes-gamma } n :: 'a))$

**using** *stieltjes-gamma-real-limit-form*[of *n*] **by** (*intro tendsto-of-real*) (*auto simp: add-ac*)

**thus** *?thesis by simp*

**qed**

**lemma** *stieltjes-gamma-real-altdef*:

$(\text{stieltjes-gamma } n :: \text{real}) =$   
 $\lim (\lambda m. (\sum k = 1..m. \ln (\text{real } k) \wedge n / \text{real } k) -$   
 $\ln (\text{real } m) \wedge (n+1) / \text{real } (n+1))$   
**by** (*rule sym, rule limI, rule stieltjes-gamma-real-limit-form*)

### 3.2 Proof of the Laurent expansion

We shall follow the proof by Briggs and Chowla [2], which examines the entire function  $g(s) = (2^{1-s} - 1)\zeta(s)$ . They determine the value of  $g^{(k)}(1)$  in two different ways: First by the Dirichlet series of  $g$  and then by its power series expansion around 1. We shall do the same here.

**context**

**fixes**  $g$  **and**  $G1\ G2\ G2'\ G :: \text{complex fps}$  **and**  $A :: \text{nat} \Rightarrow \text{complex}$

```

defines  $g \equiv \text{perzeta } (1 / 2)$ 
defines  $G1 \equiv \text{fps-shift } 1 \text{ (fps-exp } (-\ln 2 :: \text{complex}) - 1)$ 
defines  $G2 \equiv \text{fps-expansion } (\lambda s. (s - 1) * \text{pre-zeta } 1 s + 1) 1$ 
defines  $G2' \equiv \text{fps-expansion } (\text{pre-zeta } 1) 1$ 
defines  $G \equiv G1 * G2$ 
defines  $A \equiv \text{fps-nth } G2$ 
begin

```

$G1$ ,  $G2$ ,  $G2'$ , and  $G2$  are the formal power series expansions of functions around  $s = 1$  of the entire functions

- $(2^{1-s} - 1)/(s - 1)$ ,
- $(s - 1)\zeta(s)$ ,
- $\zeta(s) - \frac{1}{s-1}$ ,
- $(2^{1-s} - 1)\zeta(s)$ ,

respectively.

Our goal is to determine the coefficients of  $G2'$ , and we shall do so by determining the coefficients of  $G2$  (which are the same, but shifted by 1). This in turn will be done by determining the coefficients of  $G = G1 * G2$ . Note that  $(2^{1-s} - 1)\zeta(s)$  is written as *perzeta* (1 / 2) in Isabelle (using the periodic  $\zeta$  function) and the analytic continuation of  $\zeta(s) - \frac{1}{s-1}$  is written as *pre-zeta* 1  $s$  (*pre-zeta* is an artefact from the definition of *zeta*, which comes in useful here).

**lemma** *stieltjes-gamma-aux1*:  $(\lambda n. (-1)^{\sim(n+1)} * \ln(n+1)^{\sim k} / (n+1)) \text{ sums } ((-1)^{\sim k} * (\text{deriv } \sim k) g 1)$

**proof** –

```

define  $H$  where  $H = \text{fds-perzeta } (1 / 2)$ 
have conv: conv-abscissa  $H < 1$  unfolding  $H\text{-def}$ 
  by (rule le-less-trans[OF conv-abscissa-perzeta']) (use fraction-not-in-ints[of 2 1] in auto)
have [simp]: eval-fds  $H s = g s$  if  $\text{Re } s > 0$  for  $s$ 
  unfolding  $H\text{-def } g\text{-def}$  using fraction-not-in-ints[of 2 1] that
  by (subst perzeta-altdef2) auto
have ev: eventually  $(\lambda s. s \in \{s. \text{Re } s > 0\})$  (nhds 1)
  by (intro eventually-nhds-in-open open-halfspace-Re-gt) auto
have [simp]:  $(\text{deriv } \sim k) (\text{eval-fds } H) 1 = (\text{deriv } \sim k) g 1$ 
  by (intro higher-deriv-cong-ev eventually-mono[OF ev]) auto

have fds-converges  $((\text{fds-deriv } \sim k) H) 1$ 
  by (intro fds-converges le-less-trans[OF conv-abscissa-higher-deriv-le])
    (use conv in <simp add: one-ereal-def>)
hence  $(\lambda n. \text{fds-nth } ((\text{fds-deriv } \sim k) H) (n+1) / \text{real } (n+1)) \text{ sums eval-fds } ((\text{fds-deriv } \sim k) H) 1$ 
  by (simp add: fds-converges-altdef)

```

**also have**  $eval-fds ((fds-deriv \hat{\sim} k) H) 1 = (deriv \hat{\sim} k) (eval-fds H) 1$   
**using**  $conv$  **by**  $(intro\ eval-fds-higher-deriv)$   $(auto\ simp: one-ereal-def)$   
**also have**  $(\lambda n. fds-nth ((fds-deriv \hat{\sim} k) H) (n+1) / real (n+1)) =$   
 $(\lambda n. (-1)^{\hat{\sim} k} * (-1)^{\hat{\sim}(n+1)} * \ln (real (n+1))^{\hat{\sim} k} / (n+1))$   
**by**  $(auto\ simp: fds-nth-higher-deriv\ algebra-simps\ H-def\ fds-perzeta-one-half\ Ln-Reals-eq)$   
**finally have**  $(\lambda n. (-1)^{\hat{\sim} k} * complex-of-real ((-1)^{\hat{\sim}(n+1)} * \ln (real (n+1))^{\hat{\sim} k} / real (n+1)))\ sums$   
 $((deriv \hat{\sim} k) g 1)$  **by**  $(simp\ add: algebra-simps)$   
  
**hence**  $(\lambda n. (-1)^{\hat{\sim} k} * ((-1)^{\hat{\sim} k} * complex-of-real ((-1)^{\hat{\sim}(n+1)} * \ln (real (n+1))^{\hat{\sim} k} / real (n+1)))\ sums$   
 $((-1)^{\hat{\sim} k} * (deriv \hat{\sim} k) g 1)$  **by**  $(intro\ sums-mult)$   
**also have**  $(\lambda n. (-1)^{\hat{\sim} k} * ((-1)^{\hat{\sim} k} * complex-of-real ((-1)^{\hat{\sim}(n+1)} * \ln (real (n+1))^{\hat{\sim} k} / real (n+1)))) =$   
 $(\lambda n. complex-of-real ((-1)^{\hat{\sim}(n+1)} * \ln (real (n+1))^{\hat{\sim} k} / real (n+1)))$   
**by**  $(intro\ ext)\ auto$   
**finally show**  $?thesis$  .  
**qed**

**lemma**  $stieltjes-gamma-aux2: (deriv \hat{\sim} k) g 1 = fact k * fps-nth G k$

**and**  $stieltjes-gamma-aux3: G2 = fps-X * G2' + 1$

**proof** –

**have**  $[simp]: fps-conv-radius G1 = \infty$

**using**  $fps-conv-radius-diff[of\ fps-exp (-Ln 2) 1]$  **by**  $(simp\ add: G1-def)$

**have**  $fps-conv-radius G2 \geq \infty$

**unfolding**  $G2-def$  **by**  $(intro\ conv-radius-fps-expansion\ holomorphic-intros)\ auto$

**hence**  $[simp]: fps-conv-radius G2 = \infty$

**by**  $simp$

**have**  $fps-conv-radius G2' \geq \infty$

**unfolding**  $G2'-def$  **by**  $(intro\ conv-radius-fps-expansion\ holomorphic-intros)$

$auto$

**hence**  $[simp]: fps-conv-radius G2' = \infty$

**by**  $simp$

**have**  $[simp]: fps-conv-radius G = \infty$

**using**  $fps-conv-radius-mult[of\ G1\ G2]$  **by**  $(simp\ add: G-def)$

**have**  $eval-G1: eval-fps G1 (s - 1) =$

$(if\ s = 1\ then\ -\ln 2\ else\ (2\ powr (1 - s) - 1) / (s - 1))$  **for**  $s$

**unfolding**  $G1-def$  **using**  $fps-conv-radius-diff[of\ fps-exp (-Ln 2) 1]$

**by**  $(subst\ eval-fps-shift)$

$(auto\ intro!: subdegree-geI\ simp: eval-fps-diff\ powr-def\ exp-diff\ exp-minus$

$algebra-simps)$

**have**  $eval-G2: eval-fps G2 (s - 1) = (s - 1) * pre-zeta 1 s + 1$  **for**  $s$

**unfolding**  $G2-def$  **by**  $(subst\ eval-fps-expansion[where\ r = \infty])\ (auto\ intro!:$

$holomorphic-intros)$

**have**  $eval-G: eval-fps G (s - 1) = g s$  **for**  $s$

**unfolding**  $G-def$  **by**  $(simp\ add: eval-fps-mult\ eval-G1\ eval-G2\ g-def\ perzeta-one-half-left')$

**have**  $eval-G': eval-fps G s = g (1 + s)$  **for**  $s$



**using**  $\text{eval-G}$ [of  $s + 1$ ] **by** ( $\text{simp add: add-ac}$ )  
**have**  $\text{eval-G}2'$ :  $\text{eval-fps } G2' (s - 1) = \text{pre-zeta } 1 s$  **for**  $s$   
**unfolding**  $G2'$ -def **by** ( $\text{intro eval-fps-expansion}[\text{where } r = \infty]$ ) ( $\text{auto intro!:$   
 $\text{holomorphic-intros}$ )

**show**  $G2 = \text{fps-X} * G2' + 1$   
**proof** ( $\text{intro eval-fps-eqD always-eventually allI}$ )  
**have**  $*$ :  $\text{fps-conv-radius} (\text{fps-X} * G2') = \infty$   
**using**  $\text{fps-conv-radius-mult}$ [of  $\text{fps-X } G2'$ ] **by**  $\text{simp}$   
**from**  $*$  **show**  $\text{fps-conv-radius} (\text{fps-X} * G2' + 1) > 0$   
**using**  $\text{fps-conv-radius-add}$ [of  $\text{fps-X} * G2' 1$ ] **by**  $\text{auto}$   
**show**  $\text{eval-fps } G2 s = \text{eval-fps} (\text{fps-X} * G2' + 1) s$  **for**  $s$   
**using**  $*$   $\text{eval-G}$ [of  $1 + s$ ]  $\text{eval-G}2'$ [of  $1 + s$ ]  
**by** ( $\text{simp add: eval-fps-add eval-fps-mult}$ )  
**qed**  $\text{auto}$

**have**  $G = \text{fps-expansion } g 1$   
**proof** ( $\text{rule eval-fps-eqD}$ )  
**have**  $\text{fps-conv-radius} (\text{fps-expansion } g 1) \geq \infty$   
**using**  $\text{fraction-not-in-ints}$ [of  $2 1$ ]  
**by** ( $\text{intro conv-radius-fps-expansion}$ ) ( $\text{auto intro!:$   $\text{holomorphic-intros simp:}$   
 $g\text{-def}$ )  
**thus**  $\text{fps-conv-radius} (\text{fps-expansion } g 1) > 0$  **by**  $\text{simp}$   
**next**  
**have**  $\text{eval-fps} (\text{fps-expansion } g 1) z = g (1 + z)$  **for**  $z$   
**using**  $\text{fraction-not-in-ints}$ [of  $2 1$ ]  
**by** ( $\text{subst eval-fps-expansion}'[\text{where } r = \infty]$ ) ( $\text{auto simp: } g\text{-def intro!:$   $\text{holo-}$   
 $\text{morphic-intros}$ )  
**thus**  $\text{eventually } (\lambda z. \text{eval-fps } G z = \text{eval-fps} (\text{fps-expansion } g 1) z)$  ( $\text{nhds } 0$ )  
**by** ( $\text{simp add: eval-G}'$ )  
**qed**  $\text{auto}$   
**thus** ( $\text{deriv } \sim k$ )  $g 1 = \text{fact } k * \text{fps-nth } G k$   
**by** ( $\text{simp add: fps-eq-iff fps-expansion-def}$ )  
**qed**

**lemma**  $\text{stieltjes-gamma-aux4}$ :  $\text{fps-nth } G k = (\sum_{i=1..k+1}. (-\ln 2)^{\sim i} * A (k-(i-1))) / \text{fact } i$

**proof** –  
**have**  $\text{fps-nth } G k = (\sum_{i \leq k}. \text{fps-nth } G1 i * A (k - i))$   
**unfolding**  $G\text{-def fps-mult-nth } A\text{-def}$  **by** ( $\text{intro sum.cong}$ )  $\text{auto}$   
**also have**  $\dots = (\sum_{i \leq k}. (-\ln 2)^{\sim(i+1)} * A (k - i) / \text{fact } (i+1))$   
**by** ( $\text{simp add: } G1\text{-def algebra-simps}$ )  
**also have**  $\dots = (\sum_{i=1..k+1}. (-\ln 2)^{\sim i} * A (k-(i-1)) / \text{fact } i)$   
**by** ( $\text{intro sum.reindex-bij-witness}$ [of  $-\lambda i. i-1 \text{ Suc}$ ]) ( $\text{auto simp: Suc-diff-Suc}$ )  
**finally show**  $?thesis$  .  
**qed**

**lemma**  $\text{stieltjes-gamma-aux5}$ :  $(\sum_{t < k}. (k \text{ choose } t) * \ln 2^{\sim (k - t)} * \text{stieltjes-gamma } t)$  –

$\ln 2^{\wedge(k+1)} / \text{of-nat } (k+1) = (-1)^{\wedge k} * (\text{deriv } \sim k) g 1$

**proof** –

**define**  $h$  **where**  $h = (\lambda k x. (\sum n=1..x. \ln(\text{real } n)^{\wedge k} / \text{real } n) - \ln(\text{real } x)^{\wedge(k+1)} / \text{real}(k+1) - \text{stieltjes-gamma } k)$

**have**  $h\text{-eq}$ :  $(\sum n=1..x. \ln n^{\wedge k} / n) = \ln x^{\wedge(k+1)} / \text{real } (k+1) + \text{stieltjes-gamma } k + h k x$

**for**  $k x :: \text{nat}$  **by** (*simp add: h-def*)

**define**  $h'$  **where**  $h' = (\lambda x. \sum t=0..k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * h t x)$

**define**  $S1$  **where**  $S1 = (\lambda x. (\sum t=0..k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln x^{\wedge(t+1)} / (t+1)))$

**define**  $S2$  **where**  $S2 = (\lambda x. (\sum t=0..k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln x^{\wedge(t+1)} / (k+1)))$

**have** [*THEN filterlim-compose, tendsto-intros*]:  $h t \longrightarrow 0$  **for**  $t$

**using** *tendsto-diff[OF stieltjes-gamma-real-limit-form[of t] tendsto-const[of stieltjes-gamma t]]*

**by** (*simp add: h-def*)

**have**  $eq$ :  $(\sum n=1..2 * x. (-1)^{\wedge(n+1)} * \ln n^{\wedge k} / n) = \ln 2^{\wedge(k+1)} / \text{real } (k+1) - (\sum t < k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \text{stieltjes-gamma } t) + h k (2*x) - h' x$

**(is ?lhs x = ?rhs x) if**  $x > 0$  **for**  $x :: \text{nat}$

**proof** –

**have**  $2 * (\sum n=1..x. \ln (2*n)^{\wedge k} / (2*n)) = (\sum n=1..x. \sum t=0..k. 1/n * (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln n^{\wedge t})$

**unfolding** *sum-distrib-left*

**proof** (*rule sum.cong*)

**fix**  $n :: \text{nat}$  **assume**  $n: n \in \{1..x\}$

**have**  $2 * (\ln (2*n)^{\wedge k} / (2*n)) = 1/n * (\ln n + \ln 2)^{\wedge k}$

**using**  $n$  **by** (*simp add: ln-mult add-ac*)

**also have**  $(\ln n + \ln 2)^{\wedge k} = (\sum t=0..k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln n^{\wedge t})$

**by** (*subst binomial-ring, rule sum.cong*) *auto*

**also have**  $1/n * \dots = (\sum t=0..k. 1/n * (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln n^{\wedge t})$

**by** (*subst sum-distrib-left*) (*simp add: mult-ac*)

**finally show**  $2 * (\ln (2*n)^{\wedge k} / (2*n)) = \dots$

**qed** *auto*

**also have**  $\dots = (\sum t=0..k. \sum n=1..x. 1/n * (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln n^{\wedge t})$

**by** (*rule sum.swap*)

**also have**  $\dots = (\sum t=0..k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * (\ln x^{\wedge(t+1)} / (t+1) + \text{stieltjes-gamma } t + h t x))$

**proof** (*rule sum.cong*)

**fix**  $t :: \text{nat}$  **assume**  $t: t \in \{0..k\}$

**have**  $(\sum n=1..x. 1/n * (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \ln n^{\wedge t}) = (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * (\sum n=1..x. \ln n^{\wedge t} / n)$

**by** (*subst sum-distrib-left*) (*simp add: mult-ac*)

**also have**  $(\sum_{n=1..x}. \ln n^{\wedge} t / n) = \ln x^{\wedge}(t+1) / (t+1) + \text{stieltjes-gamma } t + h \ t \ x$   
**using** *h-eq[of t] by simp*  
**finally show**  $(\sum_{n=1..x}. 1/n * (k \text{ choose } t) * \ln 2^{\wedge}(k-t) * \ln n^{\wedge} t) =$   
 $(k \text{ choose } t) * \ln 2^{\wedge}(k-t) * \dots$   
**qed** *simp-all*  
**also have**  $\dots = (\sum_{t=0..k}. (k \text{ choose } t) / (t+1) * \ln 2^{\wedge}(k-t) * \ln x^{\wedge}(t+1)) +$   
 $(\sum_{t=0..k}. (k \text{ choose } t) * \ln 2^{\wedge}(k-t) * \text{stieltjes-gamma } t) + h' \ x$   
**by** *(simp add: ring-distrib sum.distrib h'-def)*  
**also have**  $(\sum_{t=0..k}. (k \text{ choose } t) / (t+1) * \ln 2^{\wedge}(k-t) * \ln x^{\wedge}(t+1))$   
 $=$   
 $(\sum_{t=0..k}. (\text{Suc } k \text{ choose } \text{Suc } t) / (k+1) * \ln 2^{\wedge}(k-t) * \ln x^{\wedge}(t+1))$   
**proof** *(intro sum.cong refl, goal-cases)*  
**case**  $(1 \ t)$   
**have** *of-nat (k choose t) \* (of-nat (k+1) :: real) = of-nat ((k choose t) \* (k+1))*  
**by** *(simp only: of-nat-mult)*  
**also have**  $(k \text{ choose } t) * (k+1) = (\text{Suc } k \text{ choose } \text{Suc } t) * (t+1)$   
**using** *Suc-times-binomial-eq[of k t] by (simp add: algebra-simps)*  
**also have** *of-nat ... = of-nat (Suc k choose Suc t) \* (of-nat (t+1) :: real)*  
**by** *(simp only: of-nat-mult)*  
**finally have**  $*$ : *of-nat (k choose t) / of-nat (t+1) =*  
 $(\text{of-nat } (\text{Suc } k \text{ choose } \text{Suc } t) / (k+1) :: \text{real})$   
**by** *(simp add: divide-simps flip: of-nat-Suc del: binomial-Suc-Suc)*  
**show** *?case by (simp only: \*)*  
**qed**  
**also have**  $\dots = (\sum_{t=1..Suc \ k}. (\text{Suc } k \text{ choose } t) / (k+1) * \ln 2^{\wedge}(\text{Suc } k - t) * \ln x^{\wedge} t)$   
**by** *(intro sum.reindex-bij-witness[of - \lambda t. t-1 Suc]) auto*  
**also have**  $\{1..Suc \ k\} = \{..Suc \ k\} - \{0\}$  **by** *auto*  
**also have**  $(\sum_{t \in \dots}. (\text{Suc } k \text{ choose } t) / (k+1) * \ln 2^{\wedge}(\text{Suc } k - t) * \ln x^{\wedge} t) =$   
 $(\sum_{t \leq \text{Suc } k}. (\text{Suc } k \text{ choose } t) / (k+1) * \ln 2^{\wedge}(\text{Suc } k - t) * \ln x^{\wedge} t) -$   
 $\ln 2^{\wedge} \text{Suc } k / (k+1)$   
**by** *(subst sum-diff1) auto*  
**also have**  $(\sum_{t \leq \text{Suc } k}. (\text{Suc } k \text{ choose } t) / (k+1) * \ln 2^{\wedge}(\text{Suc } k - t) * \ln x^{\wedge} t) =$   
 $(\ln x + \ln 2)^{\wedge} \text{Suc } k / (k+1)$   
**unfolding** *binomial-ring by (subst sum-divide-distrib) (auto simp: algebra-simps)*  
**also have**  $\ln x + \ln 2 = \ln (2 * x)$   
**using**  $\langle x > 0 \rangle$  **by** *(simp add: ln-mult)*  
**finally have** *eq1: 2 \* (\sum\_{n=1..x}. \ln (real (2\*n))^k / real (2\*n)) =*  
 $\ln (real (2*x))^{\wedge}(k+1) / real (k+1) - \ln 2^{\wedge}(k+1) / real (k+1)$   
 $+$   
 $(\sum_{t=0..k}. (k \text{ choose } t) * \ln 2^{\wedge}(k-t) * \text{stieltjes-gamma } t) +$

$h' x$   
**by** (*simp add: algebra-simps*)

**have**  $eq2: (\sum_{n=1..2*x} \ln n^k / n) = \ln (\text{real } (2*x))^{(k+1)} / \text{real } (k+1)$   
 $+ \text{stieltjes-gamma } k + h k (2*x)$   
**by** (*simp only: h-eq*)

**have**  $(\sum_{n=1..2*x} (-1)^{(n+1)} * \ln n^k / n) =$   
 $(\sum_{n=1..2*x} \ln n^k / n - 2 * (\text{if even } n \text{ then } \ln n^k / n \text{ else } 0))$   
**by** (*intro sum.cong*) *auto*

**also have**  $\dots = (\sum_{n=1..2*x} \ln n^k / n) -$   
 $2 * (\sum_{n=1..2*x} \text{if even } n \text{ then } \ln n^k / n \text{ else } 0)$   
**by** (*simp only: sum-subtractf sum-distrib-left*)

**also have**  $(\sum_{n=1..2*x} \text{if even } n \text{ then } \ln n^k / n \text{ else } 0) =$   
 $(\sum_{n \mid n \in \{1..2*x\} \wedge \text{even } n} \ln n^k / n)$   
**by** (*intro sum.mono-neutral-cong-right*) *auto*

**also have**  $\dots = (\sum_{n=1..x} \ln (\text{real } (2*n))^k / \text{real } (2*n))$   
**by** (*intro sum.reindex-bij-witness[of - \lambda n. 2\*n \lambda n. n div 2]*) *auto*

**also have**  $(\sum_{n=1..2*x} \ln n^k / n) - 2 * \dots =$   
 $\ln 2^{(k+1)} / \text{real } (k+1) -$   
 $((\sum_{t=0..k} (k \text{ choose } t) * \ln 2^{(k-t)} * \text{stieltjes-gamma } t) -$   
 $\text{stieltjes-gamma } k) +$   
 $h k (2*x) - h' x$   
**using** *arg-cong2[OF eq1 eq2, of (-)]* **by** *simp*

**also have**  $\{0..k\} = \text{insert } k \{..<k\}$  **by** *auto*

**also have**  $(\sum_{t \in \dots} (k \text{ choose } t) * \ln 2^{(k-t)} * \text{stieltjes-gamma } t) - \text{stieltjes-gamma } k =$   
 $(\sum_{t < k} (k \text{ choose } t) * \ln 2^{(k-t)} * \text{stieltjes-gamma } t)$   
**by** (*subst sum.insert*) *auto*

**finally show** *?thesis* .  
**qed**

**have**  $?rhs \longrightarrow \ln 2^{(k+1)} / \text{real } (k+1) -$   
 $(\sum_{t < k} (k \text{ choose } t) * \ln 2^{(k-t)} * \text{stieltjes-gamma } t)$   
**unfolding** *h'-def* **by** (*rule tendsto-eq-intros refl mult-nat-left-at-top filter-lim-ident | simp*) $+$

**moreover have** *eventually*  $(\lambda x. ?rhs x = ?lhs x)$  *sequentially*  
**using** *eventually-gt-at-top[of 0]* **by** *eventually-elim (simp only: eq)*

**ultimately have**  $*: ?lhs \longrightarrow \ln 2^{(k+1)} / \text{real } (k+1) -$   
 $(\sum_{t < k} (k \text{ choose } t) * \ln 2^{(k-t)} * \text{stieltjes-gamma } t)$   
**by** (*rule Lim-transform-eventually*)

**also have**  $(\lambda x. \sum_{n=1..2*x} (-1)^{(n+1)} * \ln (\text{real } n)^k / \text{real } n) =$   
 $(\lambda x. \sum_{n < 2*x} -((-1)^{(n+1)} * \ln (\text{real } (n+1))^k / \text{real } (n+1)))$   
**by** (*intro ext sum.reindex-bij-witness[of - Suc \lambda n. n - 1]*) (*auto simp: power-diff*)

**also have**  $\dots = (\lambda x. -(\sum_{n < 2*x} ((-1)^{(n+1)} * \ln (\text{real } (n+1))^k / \text{real } (n+1))))$   
**by** (*subst sum-negf*) *auto*

**finally have**  $*: \dots \longrightarrow (\ln 2^{(k+1)} / \text{real } (k+1) -$   
 $(\sum_{t < k} (k \text{ choose } t) * \ln 2^{(k-t)} * \text{stieltjes-gamma } t))$  .

**have** *lim1*:  $(\lambda x. (\sum n < 2 * x. \text{complex-of-real } ((-1)^{\wedge(n+1)} * \ln (\text{real } (n+1)))^{\wedge k} / \text{real } (n+1))))$   
 $\longrightarrow -(\ln 2^{\wedge(k+1)} / \text{of-nat } (k+1) -$   
 $(\sum t < k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \text{stieltjes-gamma } t))$   
**(is** *?lhs'*  $\longrightarrow -)$   
**using** *tendsto-of-real*[*OF tendsto-minus*[*OF \**], **where** *?'a = complex*]  
**by** (*simp add: Ln-Reals-eq*)

**moreover have** *?lhs'*  $\longrightarrow ((-1)^{\wedge k} * (\text{deriv } \sim k) g 1)$

**proof** –

**have** \*\*: *filterlim*  $(\lambda n::\text{nat}. 2 * n)$  *sequentially sequentially* **by** *real-asymp*

**have**  $(\lambda x. (\sum n < 2 * x. \text{complex-of-real } ((-1)^{\wedge(n+1)} * \ln (\text{real } (n+1)))^{\wedge k} / \text{real } (n+1))))$   
 $\longrightarrow ((-1)^{\wedge k} * (\text{deriv } \sim k) g 1)$

**by** (*rule filterlim-compose*[*OF - \*\**]) (*use stieltjes-gamma-aux1 in <simp add: sums-def>*)

**thus** *?thesis* .

**qed**

**ultimately have**  $-(\ln 2^{\wedge(k+1)} / \text{of-nat } (k+1) -$   
 $(\sum t < k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \text{stieltjes-gamma } t)) =$   
 $(-1)^{\wedge k} * (\text{deriv } \sim k) g 1$

**by** (*rule LIMSEQ-unique*)

**thus** *?thesis* **by** (*simp add: Ln-Reals-eq*)

**qed**

**lemma** *stieltjes-gamma-aux6*:  $(\sum t < k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \text{stieltjes-gamma } t) -$

$\ln 2^{\wedge(k+1)} / \text{of-nat } (k+1) =$   
 $(-1)^{\wedge k} * \text{fact } k * (\sum i=1..k+1. (-\ln 2)^{\wedge i} * A(k-(i-1)) / \text{fact } i)$

**proof** –

**have**  $(\sum t < k. (k \text{ choose } t) * \ln 2^{\wedge(k-t)} * \text{stieltjes-gamma } t) -$

$\ln 2^{\wedge(k+1)} / \text{of-nat } (k+1) = (-1)^{\wedge k} * (\text{deriv } \sim k) g 1$

**using** *stieltjes-gamma-aux5*[*of k*] .

**also have**  $(\text{deriv } \sim k) g 1 = \text{fact } k * \text{fps-nth } G k$

**by** (*rule stieltjes-gamma-aux2*)

**also have**  $\text{fps-nth } G k = (\sum i=1..k+1. (-\ln 2)^{\wedge i} * A(k-(i-1)) / \text{fact } i)$

**by** (*rule stieltjes-gamma-aux4*)

**finally show** *?thesis* **by** (*simp add: mult-ac*)

**qed**

**theorem** *higher-deriv-pre-zeta-1-1*:  $(\text{deriv } \sim k) (\text{pre-zeta } 1) 1 = (-1)^{\wedge k} * \text{stieltjes-gamma } k$

**proof** –

**have** *eq*:  $A k = (\text{if } k = 0 \text{ then } 1 \text{ else } (-1)^{\wedge(k+1)} * \text{stieltjes-gamma } (k-1) / \text{fact } (k-1))$  **for** *k*

**proof** (*induction k rule: less-induct*)

**case** (*less k*)

```

show ?case
proof (cases k = 0)
  case True
  with stieltjes-gamma-aux6[of 0] show ?thesis by simp
next
  case False
  have k * Ln 2 * stieltjes-gamma (k - 1) +
    (∑ t < k - 1. (k choose t) * Ln 2 ^ (k - t) * stieltjes-gamma t) =
    (∑ t ∈ insert (k - 1) {..<k-1}. (k choose t) * Ln 2 ^ (k - t) *
stieltjes-gamma t)
    using False by (subst sum.insert) auto
  also have insert (k - 1) {..<k-1} = {..<k} using False by auto
  also have (∑ t < k. of-nat (k choose t) * Ln 2 ^ (k - t) * stieltjes-gamma t)
=
  Ln 2 ^ (k + 1) / of-nat (k + 1) +
  (- 1) ^ k * fact k * (∑ i = 1..k + 1. (- Ln 2) ^ i * A (k - (i - 1))
/ fact i)
  using stieltjes-gamma-aux6[of k] by (simp add: algebra-simps)
  also have {1..k+1} = {1,k+1} ∪ {2..k} by auto
  also have (- 1) ^ k * fact k * (∑ i ∈ ... (- Ln 2) ^ i * A (k - (i - 1)) /
fact i) =
  (∑ i = 2..k. (- 1) ^ k * fact k * (- Ln 2) ^ i * A (k - (i - 1)) / fact
i)
  - Ln 2 * A k * (- 1) ^ k * fact k +
  (- Ln 2) ^ (k+1) * A 0 / fact (k+1) * (- 1) ^ k * fact k
  using False by (subst sum.union-disjoint)
  (auto simp: algebra-simps sum-distrib-left sum-distrib-right)
  also have (∑ i = 2..k. (- 1) ^ k * fact k * (- Ln 2) ^ i * A (k - (i - 1)) / fact
i) =
  (∑ i < k - 1. (k choose i) * Ln 2 ^ (k - i) * stieltjes-gamma i)
  using False
  by (intro sum.reindex-bij-witness[of - λi. k - i λi. k - i])
  (auto simp: binomial-fact Suc-diff-le less field-simps power-neg-one-If)
  finally have k * Ln 2 * stieltjes-gamma (k - 1) =
  (- 1) ^ (k+1) * fact k * Ln 2 * A k
  using False by (simp add: less power-minus')
  also have ... * (- 1) ^ (k+1) / fact k / Ln 2 = A k
  by simp
  also have k * Ln 2 * stieltjes-gamma (k - 1) * (- 1) ^ (k+1) / fact k / Ln
2 =
  (- 1) ^ (k+1) * stieltjes-gamma (k - 1) / fact (k - 1)
  using False by (simp add: field-simps fact-reduce)
  finally have A k = (- 1) ^ (k + 1) * stieltjes-gamma (k - 1) / fact (k -
1) ..
  thus ?thesis using False by simp
qed
qed

have fps-nth G2' k = fps-nth G2 (Suc k)

```

by (simp add: stieltjes-gamma-aux3)  
 also have ... = A (Suc k)  
 by (simp add: A-def)  
 also have ... =  $(-1)^{\wedge k} * \text{stieltjes-gamma } k / \text{fact } k$   
 by (simp add: eq)  
 finally show (deriv  $\widehat{\wedge} k$ ) (pre-zeta 1) 1 =  $(-1)^{\wedge k} * \text{stieltjes-gamma } k$   
 by (simp add: G2'-def fps-eq-iff fps-expansion-def)  
 qed

**corollary** pre-zeta-1-1 [simp]: pre-zeta 1 1 = euler-mascheroni  
 using higher-deriv-pre-zeta-1-1[of 0] by simp

**corollary** zeta-minus-pole-limit:  $(\lambda s. \text{zeta } s - 1 / (s - 1)) -1 \rightarrow \text{euler-mascheroni}$   
**proof** (rule Lim-transform-eventually)

show eventually  $(\lambda s. \text{pre-zeta } 1 s = \text{zeta } s - 1 / (s - 1))$  (at 1)  
 by (auto simp: zeta-minus-pole-eq [symmetric] eventually-at-filter)  
 have isCont (pre-zeta 1) 1  
 by (intro continuous-intros) auto  
 thus pre-zeta 1  $-1 \rightarrow \text{euler-mascheroni}$   
 by (simp add: isCont-def)  
 qed

**corollary** fps-expansion-pre-zeta-1-1:

fps-expansion (pre-zeta 1) 1 = Abs-fps  $(\lambda n. (-1)^{\wedge n} * \text{stieltjes-gamma } n / \text{fact } n)$   
 by (simp add: fps-expansion-def higher-deriv-pre-zeta-1-1)

end

**definition** fps-pre-zeta-1 :: complex fps **where**

fps-pre-zeta-1 = Abs-fps  $(\lambda n. (-1)^{\wedge n} * \text{stieltjes-gamma } n / \text{fact } n)$

**lemma** pre-zeta-1-has-fps-expansion-1 [fps-expansion-intros]:

$(\lambda z. \text{pre-zeta } 1 (1 + z))$  has-fps-expansion fps-pre-zeta-1

**proof** –

have  $(\lambda z. \text{pre-zeta } 1 (1 + z))$  has-fps-expansion fps-expansion (pre-zeta 1) 1

by (intro analytic-at-imp-has-fps-expansion analytic-intros analytic-pre-zeta)

auto

also have ... = fps-pre-zeta-1

by (simp add: fps-expansion-pre-zeta-1-1 fps-pre-zeta-1-def)

finally show ?thesis .

qed

**definition** fls-zeta-1 :: complex fls **where**

fls-zeta-1 = fls-X-intpow  $(-1) + \text{fps-to-fls } \text{fps-pre-zeta-1}$

**lemma** zeta-has-laurent-expansion-1 [laurent-expansion-intros]:

$(\lambda z. \text{zeta } (1 + z))$  has-laurent-expansion fls-zeta-1

**proof** –

```

have ( $\lambda z. z \text{ powi } (-1) + \text{pre-zeta } 1 (1 + z)$ ) has-laurent-expansion fls-zeta-1
  unfolding fls-zeta-1-def
  by (intro laurent-expansion-intros fps-expansion-intros has-laurent-expansion-fps)
also have ?this  $\longleftrightarrow$  ?thesis
  by (intro has-laurent-expansion-cong)
    (auto simp: eventually-at-filter zeta-def hurwitz-zeta-def divide-inverse)
finally show ?thesis .
qed

end

```

## 4 The Hadjicostas–Chapman formula

```

theory Hadjicostas-Chapman
  imports Zeta-Laurent-Expansion
begin

```

In this section, we will derive a formula for the  $\zeta$  function that was conjectured by Hadjicostas [4] and proven shortly afterwards by Chapman [3]. The formula is:

$$\begin{aligned}
 & \int_0^1 \int_0^1 \frac{(-\ln(xy))^z (1-x)}{1-xy} \, dx \, dy \\
 &= \int_0^1 \frac{(-\ln u)^z (-\ln u + u - 1)}{1-u} \, du \\
 &= \Gamma(z+2) \left( \zeta(z+2) - \frac{1}{z+1} \right)
 \end{aligned}$$

for any  $z$  with  $\Re(z) > -2$ . In particular, setting  $z = 1$ , we can derive the following formula for the Euler–Mascheroni constant  $\gamma$ :

$$- \int_0^1 \int_0^1 \frac{1-x}{(1-xy) \ln(xy)} \, dx \, dy = \gamma$$

This formula was first proven by Sondow [7].

### 4.1 The real case

We first define the integral for real  $z > -2$ . This is then a non-negative integral, so that we can ignore the issue of integrability and use the Lebesgue integral on the extended non-negative reals

We first show the equivalence of the single-integral and the double-integral form.

```

definition Hadjicostas-nn-integral :: real  $\Rightarrow$  ennreal where
  Hadjicostas-nn-integral  $z =$ 
    set-nn-integral lborel  $\{0 < .. < 1\}$ 

```



$$(\lambda u. \text{ennreal } ((-\ln u) \text{ powr } z / (1 - u) * (-\ln u + u - 1)))$$

**definition** *Hadjicostas-integral* :: complex  $\Rightarrow$  complex **where**

*Hadjicostas-integral* z =  
(LBINT u=0..1. of-real (-ln u) powr z / of-real (1 - u) \* of-real (-ln u + u - 1))

**lemma** *Hadjicostas-nn-integral-altdef*:

*Hadjicostas-nn-integral* z =  
 $(\int^{+}(x,y) \in \{0 <..< 1\} \times \{0 <..< 1\}. ((-\ln (x*y)) \text{ powr } z * (1-x) / (1-x*y)) \partial \text{lborel})$

**proof** -

**define** f **where** f  $\equiv (\lambda u. ((-\ln u) \text{ powr } z / (1 - u) * (-\ln u + u - 1)))$

**let** ?I = Gamma (z + 2) \* (Re (zeta (z + 2)) - 1 / (z + 1))

**let** ?f =  $\lambda u. ((-\ln u) \text{ powr } z / (1 - u) * (-\ln u + u - 1))$

**define** D :: (real  $\times$  real) set **where** D =  $\{0 <..< 1\} \times \{0 <..< 1\}$

**define** D1 **where** D1 = (SIGMA x:  $\{0 <..< 1\}. \{0 <..< (x::\text{real})\}$ )

**define** D2 **where** D2 = (SIGMA u:  $\{0 <..< 1\}. \{u <..< (1::\text{real})\}$ )

**have** [measurable]: D1  $\in$  sets (lborel  $\otimes_M$  lborel)

**proof** -

**have** D1 =  $\{x \in \text{space } (\text{lborel} \otimes_M \text{lborel}). \text{snd } x > 0 \wedge \text{fst } x > \text{snd } x \wedge \text{fst } x < 1\}$

**by** (auto simp: D1-def space-pair-measure)

**also have** ...  $\in$  sets (lborel  $\otimes_M$  lborel)

**by** measurable

**finally show** ?thesis .

**qed**

**have** [measurable]: D2  $\in$  sets (lborel  $\otimes_M$  lborel)

**proof** -

**have** D2 =  $\{x \in \text{space } (\text{lborel} \otimes_M \text{lborel}). \text{fst } x > 0 \wedge \text{fst } x < \text{snd } x \wedge \text{snd } x < 1\}$

**by** (auto simp: D2-def space-pair-measure)

**also have** ...  $\in$  sets (lborel  $\otimes_M$  lborel)

**by** measurable

**finally show** ?thesis .

**qed**

**have**  $(\int^{+}(x,y) \in D. ((-\ln (x*y)) \text{ powr } z * (1-x) / (1-x*y)) \partial \text{lborel}) =$   
 $(\int^{+} x \in \{0 <..< 1\}. (\int^{+} y \in \{0 <..< 1\}. ((-\ln (x*y)) \text{ powr } z / (1-x*y) * (1-x)) \partial \text{lborel}) \partial \text{lborel})$

**unfolding** lborel-prod [symmetric] case-prod-unfold D-def

**by** (subst lborel.nn-integral-fst[symmetric])

(auto intro!: nn-integral-cong simp: indicator-def)

**also have** ... =  $(\int^{+} x \in \{0 <..< 1\}. (\int^{+} u \in \{0 <..< x\}. ((-\ln u) \text{ powr } z / (1 - u) * (1 - x) / x) \partial \text{lborel}) \partial \text{lborel})$

**proof** (rule set-nn-integral-cong)

**fix** x :: real **assume** x  $\in$  space lborel  $\cap$   $\{0 <..< 1\}$

**show**  $(\int^{+} y \in \{0 <..< 1\}. ((-\ln (x*y)) \text{ powr } z / (1-x*y) * (1-x)) \partial \text{lborel}) =$

$(\int^{+} u \in \{0 <..< x\}. ((-\ln u) \text{ powr } z / (1 - u) * (1 - x) / x) \partial \text{lborel})$

```

using x
  by (subst lborel-distr-mult'[of 1/x])
    (auto simp: nn-integral-density nn-integral-distr indicator-def field-simps
      simp flip: ennreal-mult' intro!: nn-integral-cong)
qed auto
also have ... = ( $\int^{+(x,u) \in D1. ((- \ln u) \text{ powr } z / (1 - u) * (1 - x) / x)$ 
 $\partial \text{lborel}$ )
  unfolding lborel-prod [symmetric] case-prod-unfold D-def
  by (subst lborel.nn-integral-fst[symmetric], measurable)
    (auto intro!: nn-integral-cong simp: indicator-def D1-def)
also have ... = ( $\int^{+(x,u). \text{ indicator } D2 (u,x) * ((- \ln u) \text{ powr } z / (1 - u) * (1 - x) / x)$ 
 $\partial \text{lborel}$ )
  by (intro nn-integral-cong) (auto simp: D1-def D2-def indicator-def split:
if-splits)
also have ... = ( $\int^{+u \in \{0 < .. < 1\}. (\int^{+x \in \{u < .. < 1\}. ((- \ln u) \text{ powr } z / (1 - u) * (1 - x) / x)$ 
 $\partial \text{lborel}) \partial \text{lborel}$ )
  unfolding case-prod-unfold lborel-prod [symmetric]
  by (subst lborel-pair.nn-integral-snd [symmetric], measurable)
    (auto intro!: nn-integral-cong simp: D2-def indicator-def)
also have ... = ( $\int^{+u \in \{0 < .. < 1\}. ((- \ln u) \text{ powr } z / (1 - u) * (- \ln u + u - 1))$ 
 $\partial \text{lborel}$ )
proof (intro set-nn-integral-cong refl)
  fix u :: real assume u: u  $\in$  space lborel  $\cap$  {0 < .. < 1}
  let ?F =  $\lambda x. (- \ln u) \text{ powr } z / (1 - u) * (\ln x - x)$ 
  have ( $\int^{+x \in \{u < .. < 1\}. \text{ ennreal } ((- \ln u) \text{ powr } z / (1 - u) * (1 - x) / x)$ 
 $\partial \text{lborel}$ ) =
    ( $\int^{+x \in \{u..1\}. \text{ ennreal } ((- \ln u) \text{ powr } z / (1 - u) * (1 - x) / x)$ 
 $\partial \text{lborel}$ )
  by (rule nn-integral-cong-AE, rule AE-I[of - - {u,1}])
    (auto simp: emeasure-lborel-countable indicator-def)
also have ... = ennreal (?F 1 - ?F u)
  using u by (intro nn-integral-FTC-Icc) (auto intro!: derivative-eq-intros simp:
divide-simps)
also have ?F 1 - ?F u =  $(- \ln u) \text{ powr } z / (1 - u) * (- \ln u + u - 1)$ 
  using u by (simp add: divide-simps) (simp add: algebra-simps)?
  finally show ( $\int^{+x \in \{u < .. < 1\}. ((- \ln u) \text{ powr } z / (1 - u) * (1 - x) / x)$ 
 $\partial \text{lborel}$ ) = ennreal ... .
qed
also have ... = Hadjicostas-nn-integral z
  by (simp add: Hadjicostas-nn-integral-def)
finally show ?thesis by (simp add: D-def)
qed

```

We now solve the single integral for real  $z > -1$ .

**lemma** *Hadjicostas-Chapman-aux*:

```

fixes z :: real
assumes z: z > -1
defines f  $\equiv$  ( $\lambda u. ((- \ln u) \text{ powr } z / (1 - u) * (- \ln u + u - 1))$ )
shows (f has-integral (Gamma (z + 2) * (Re (zeta (z + 2)) - 1 / (z + 1))))
{0 < .. < 1}

```

```

proof -
  let ?I = Gamma (z + 2) * (Re (zeta (z + 2)) - 1 / (z + 1))
  have nonneg: 1 ≤ x + exp (- x) if x ≥ 0 for x :: real
  proof -
    have x + (1 + (-x)) ≤ x + exp (-x)
    by (intro add-left-mono exp-ge-add-one-self)
    thus ?thesis by simp
  qed

  have eq: ((λt::real. exp (-t)) ‘ {0<..} ) = {0<.. $1$ }
  proof safe
    fix x :: real assume x: x ∈ {0<.. $1$ }
    hence x = exp (-(-ln x)) and -ln x ∈ {0<..}
    by auto
    thus x ∈ (λt. exp (-t)) ‘ {0<..} by blast
  qed auto

  have I: ((λx. x powr (z+1) / (exp x - 1) - x powr z / exp x) has-integral ?I)
  {0<..}
  proof -
    from z have z + 1 ∉ ℝ≤0
    by (auto simp: nonpos-Reals-def)
    hence z': z + 1 ∉ ℤ≤0
    using nonpos-Ints-subset-nonpos-Reals by blast
    have ((λx. x powr (z + 2 - 1) / (exp x - 1) - x powr (z + 1 - 1) / exp x)
      has-integral (Gamma (z + 2) * Re (zeta (z + 2)) - Gamma (z + 1)))
  {0<..} using z
    by (intro has-integral-diff Gamma-integral-real' Gamma-times-zeta-has-integral-real)
  auto
    also have Gamma (z + 2) * Re (zeta (z + 2)) - Gamma (z + 1) =
      Gamma (z + 2) * (Re (zeta (z + 2)) - 1 / (z + 1))
    using Gamma-plus1[of z+1] z z' by (auto simp: field-simps)
    finally show ?thesis
    by (simp add: add-ac)
  qed

  also have ?this ↔ ((λx. |-exp (-x)| * f (exp (-x))) has-integral ?I) {0<..}
    unfolding f-def
    apply (intro has-integral-cong)
    apply (auto simp: field-simps powr-add powr-def exp-add)
    apply (simp flip: exp-add)
    done
  finally have *: ((λx. |-exp (-x)| * f (exp (-x))) has-integral ?I) {0<..} .

  have ((λx. |-exp (-x)| *R f (exp (-x))) absolutely-integrable-on {0<..}) ∧
    integral {0<..} (λx. |-exp (-x)| *R f (exp (-x))) = ?I
  proof (intro conjI nonnegative-absolutely-integrable-1)
    fix x :: real assume x: x ∈ {0<..}
    thus |-exp (-x)| *R f (exp (-x)) ≥ 0
    unfolding f-def using nonneg

```

**by** (intro scaleR-nonneg-nonneg mult-nonneg-nonneg divide-nonneg-nonneg)  
*auto*  
**qed** (use \* in ⟨simp-all add: has-integral-iff⟩)

**also have** ?this  $\longleftrightarrow$   $f$  absolutely-integrable-on  $(\lambda x. \exp (- x))$  ‘ $\{0 < ..\}$ ’  $\wedge$   
 $\text{integral } ((\lambda x. \exp (- x))$  ‘ $\{0 < ..\}$ ’)  $f = ?I$

**by** (intro has-absolute-integral-change-of-variables-1')  
*(auto intro!: derivative-eq-intros inj-onI)*

**also have**  $(\lambda x::\text{real}. \exp (- x))$  ‘ $\{0 < ..\}$ ’ =  $\{0 < .. < 1\}$   
**by** (fact eq)

**finally show**  $f$  has-integral  $?I$   $\{0 < .. < 1\}$   
**by** (auto simp: has-integral-iff dest: set-lebesgue-integral-eq-integral)

**qed**

**lemma** real-zeta-ge-one-over-minus-one:

**fixes**  $z :: \text{real}$   
**assumes**  $z: z > 1$   
**shows**  $\text{Re } (\text{zeta } (\text{complex-of-real } z)) \geq 1 / (z - 1)$

**proof** –

**have** *ineq*:  $1 \leq x - \ln x$  **if**  $x \in \{0 < .. < 1\}$  **for**  $x :: \text{real}$   
**using** ln-le-minus-one[of  $x$ ] **that** **by** simp

**have** \*:  $((\lambda u. (- \ln u) \text{powr } (z - 2) * (u - \ln u - 1) / (1 - u)) \text{ has-integral } \Gamma z * (\text{Re } (\text{zeta } (\text{complex-of-real } z)) - 1 / (z - 1)))$   $\{0 < .. < 1\}$   
**using** Hadjicostas-Chapman-aux[of  $z - 2$ ]  $z$  **by** simp

**from** *ineq* **have**  $\Gamma z * (\text{Re } (\text{zeta } (\text{complex-of-real } z)) - 1 / (z - 1)) \geq 0$   
**by** (intro has-integral-nonneg[OF \*]  $z$  mult-nonneg-nonneg divide-nonneg-nonneg)

*auto*

**moreover have**  $\Gamma z > 0$   
**using** *assms* **by** (intro Gamma-real-pos) *auto*

**ultimately show**  $\text{Re } (\text{zeta } (\text{complex-of-real } z)) \geq 1 / (z - 1)$   
**by** (subst (*asm*) zero-le-mult-iff) *auto*

**qed**

We now have the formula for real  $z > -1$ .

**lemma** Hadjicostas-Chapman-formula-real:

**fixes**  $z :: \text{real}$   
**assumes**  $z: z > -1$   
**shows**  $\text{Hadjicostas-nn-integral } z = \text{ennreal } (\Gamma (z + 2) * (\text{Re } (\text{zeta } (z + 2)) - 1 / (z + 1)))$

**proof** –

**have** *nonneg*:  $1 \leq x - \ln x$  **if**  $x > 0$   $x < 1$  **for**  $x :: \text{real}$   
**proof** –

**have**  $\ln x + (1 + \ln x) \leq \ln x + \exp (\ln x)$   
**by** (intro add-left-mono exp-ge-add-one-self)

**thus** ?thesis **using** *that* **by** (simp add: exp-minus)

**qed**

**show** ?thesis

**unfolding** Hadjicostas-nn-integral-def **using** nonneg Hadjicostas-Chapman-aux[OF  $z$ ]

by (intro nn-integral-has-integral-lebesgue' mult-nonneg-nonneg divide-nonneg-nonneg)  
 auto  
 qed

## 4.2 Analyticity of the integral

To extend the formula to its full domain of validity (any complex  $z$  with  $\Re(z) > -2$ ), we will use analytic continuation. To do this, we first have to show that the integral is an analytic function of  $z$  on that domain. This is unfortunately somewhat involved, since the integral is an improper one and we first need to show uniform convergence so that we can pull the derivative inside the integral sign.

We will use the single-integral form so that we only have to deal with one integral and not two.

**context**

fixes  $f :: \text{complex} \Rightarrow \text{real} \Rightarrow \text{complex}$

defines  $f \equiv (\lambda z u. \text{of-real } (-\ln u) \text{ powr } z / \text{of-real } (1 - u) * \text{of-real } (-\ln u + u - 1))$

**begin**

**context**

fixes  $x y :: \text{real}$  and  $g1 g2 :: \text{real} \Rightarrow \text{real}$

assumes  $x > -2$

defines  $g1 \equiv (\lambda x. (-\ln x) \text{ powr } y * (x - \ln x - 1) / (1 - x))$

defines  $g2 \equiv (\lambda u. (-\ln u) \text{ powr } x * (u - \ln u - 1) / (1 - u))$

**begin**

**lemma** *integrable-bound1*:

*interval-lebesgue-integrable lborel 0 (ereal (exp (- 1))) g1*

**unfolding** *zero-ereal-def*

**proof** (*rule interval-lebesgue-integrable-bigo-left*)

**show**  $g1 \in O[\text{at-right } 0](\lambda u. u \text{ powr } (-1/2))$

**unfolding** *g1-def* **by** *real-asymp*

**show** *continuous-on*  $\{0 < .. \text{exp}(-1)\}$   $g1$

**unfolding** *g1-def* **by** (*auto intro!*: *continuous-intros*)

**have** *set-integrable lborel (einterval 0 (exp (- 1)))*  $(\lambda u. u \text{ powr } (-1/2))$

**proof** (*rule interval-integral-FTC-nonneg*)

**fix**  $u :: \text{real}$  **assume**  $u: 0 < \text{ereal } u \text{ereal } u < \text{ereal } (\text{exp } (-1))$

**show**  $((\lambda u. 2 * u \text{ powr } (1/2)) \text{ has-field-derivative } (u \text{ powr } (-1/2)))$  (*at*  $u$ )

**using**  $u$  **by** (*auto intro!*: *derivative-eq-intros simp: power2-eq-square*)

**show** *isCont*  $(\lambda u. u \text{ powr } (-1/2)) u$

**using**  $u$  **by** (*auto intro!*: *continuous-intros*)

**next**

**show**  $((\lambda u. 2 * u \text{ powr } (1/2)) \circ \text{real-of-ereal}) \longrightarrow 2 * \text{exp } (-1) \text{ powr } (1/2)$   
*(at-left (ereal (exp (- 1))))*

**unfolding** *ereal-tendsto-simps* **by** *real-asymp*

**show**  $((\lambda u. 2 * u \text{ powr } (1/2)) \circ \text{real-of-ereal}) \longrightarrow 0$  (*at-right 0*)

**unfolding** *zero-ereal-def* **unfolding** *ereal-tendsto-simps* **by** *real-asymp*

```

qed auto
thus interval-lebesgue-integrable lborel (ereal 0) (ereal (exp (- 1)))
      ( $\lambda u. u \text{ powr } (-1/2)$ )
  by (simp add: interval-lebesgue-integrable-def zero-ereal-def)
qed (auto simp add: g1-def set-borel-measurable-def)

lemma integrable-bound2:
  interval-lebesgue-integrable lborel (exp (-1)) 1 g2
  unfolding one-ereal-def
proof (rule interval-lebesgue-integrable-bigo-right)
  show  $g2 \in O[\text{at-left } 1](\lambda u. (1 - u) \text{ powr } (x + 1))$ 
  unfolding g2-def by real-asymp
  have  $\ln x \neq 0$  if  $x \in \{\exp(-1)..<1\}$  for  $x :: \text{real}$ 
  proof -
    have  $0 < \exp(-1 :: \text{real})$  by simp
    also have  $\dots \leq x$  using that by auto
    finally have  $x > 0$  .
    from that  $\langle x > 0 \rangle$  have  $\ln x < \ln 1$ 
    by (subst ln-less-cancel-iff) auto
    thus  $\ln x \neq 0$  by simp
  qed
  thus continuous-on  $\{\exp(-1)..<1\}$   $g2$ 
  unfolding g2-def by (auto intro!: continuous-intros)
  let  $?F = (\lambda u. -1 / (x + 2) * (1 - u) \text{ powr } (x + 2))$ 
  have set-integrable lborel (einterval (exp (-1)) 1) ( $\lambda u. (1 - u) \text{ powr } (x + 1)$ )
  proof (rule interval-integral-FTC-nonneg[where F = ?F])
    fix  $u :: \text{real}$  assume  $u: \text{ereal } (\exp(-1)) < \text{ereal } u \text{ereal } u < 1$ 
    show ( $?F$  has-field-derivative  $(1 - u) \text{ powr } (x + 1)$ ) (at u)
    using  $u \langle x > -2 \rangle$  by (auto intro!: derivative-eq-intros simp: one-ereal-def
add-ac)
    show isCont  $(\lambda u. (1 - u) \text{ powr } (x + 1)) u$ 
    using  $u$  by (auto intro!: continuous-intros)
  next
  show  $((\lambda u. -1 / (x + 2) * (1 - u) \text{ powr } (x + 2)) \circ \text{real-of-ereal}) \longrightarrow$ 
     $-1 / (x + 2) * (1 - \exp(-1)) \text{ powr } (x + 2)$  (at-right (ereal (exp (-
     $1))))$ )
  unfolding ereal-tendsto-simps by real-asymp
  show  $((\lambda u. -1 / (x + 2) * (1 - u) \text{ powr } (x + 2)) \circ \text{real-of-ereal}) \longrightarrow 0$ 
    (at-left 1)
  unfolding one-ereal-def unfolding ereal-tendsto-simps
  using  $\langle x > -2 \rangle$  by real-asymp
qed auto
  thus interval-lebesgue-integrable lborel (ereal (exp (- 1)))
    (ereal 1)  $(\lambda u. (1 - u) \text{ powr } (x + 1))$ 
  by (simp add: interval-lebesgue-integrable-def one-ereal-def)
qed (auto simp add: g2-def set-borel-measurable-def)

```

```

lemma bound2:
  norm (f z u) ≤ g2 u if  $z: \text{Re } z \in \{x..y\}$  and  $u: u \in \{\exp(-1)..<1\}$  for  $z u$ 

```

**proof** –  
**have**  $0 < \exp(-1::\text{real})$  **by** *simp*  
**also have**  $\dots \leq u$  **using**  $u$  **by** (*simp add: einterval-def*)  
**finally have**  $u > 0$  .

**from**  $u \langle u > 0 \rangle$  **have** *ln-u: ln u > ln (exp (-1))*  
**by** (*subst ln-less-cancel-iff*) (*auto simp: einterval-def*)  
**from**  $z \ u \langle u > 0 \rangle$  **have**  $\text{norm } (f \ z \ u) = (- \ \ln \ u) \ \text{powr } \text{Re } z * |u - \ln u - 1| / (1 - u)$   
**unfolding** *f-def norm-mult norm-divide norm-of-real*  
**by** (*simp add: norm-powr-real-powr einterval-def*)  
**also have**  $|u - \ln u - 1| = u - \ln u - 1$   
**using**  $u \langle u > 0 \rangle$  *ln-add-one-self-le-self2*[*of u - 1*] **by** (*simp add: einterval-def*)  
**also have**  $(- \ \ln \ u) \ \text{powr } \text{Re } z * (u - \ln u - 1) / (1 - u) \leq (- \ \ln \ u) \ \text{powr } x * (u - \ln u - 1) / (1 - u)$   
**using**  $z \ u \langle u > 0 \rangle$  *ln-u ln-add-one-self-le-self2*[*of u - 1*]  
**by** (*intro mult-right-mono divide-right-mono powr-mono*<sup>^</sup>) (*auto simp: einterval-def*)  
**finally show**  $\text{norm } (f \ z \ u) \leq g^2 \ u$  **by** (*simp add: g2-def*)  
**qed**

**lemma** *integrable2-aux: interval-lebesgue-integrable lborel (exp (-1)) 1 (f z)*  
**if**  $z: \text{Re } z \in \{x..y\}$  **for**  $z$

**proof** –  
**have** *set-integrable lborel {exp (-1)<..  
**proof** (*rule set-integrable-bound[OF - - AE-I2[OF impI]]*)  
**fix**  $u :: \text{real}$  **assume**  $u \in \{exp (-1)<..  
**hence**  $\text{norm } (f \ z \ u) \leq g^2 \ u$  **using**  $z$  **by** (*intro bound2*) *auto*  
**also have**  $\dots \leq \text{norm } (g^2 \ u)$  **by** *simp*  
**finally show**  $\text{norm } (f \ z \ u) \leq \text{norm } (g^2 \ u)$  .  
**qed** (*use integrable-bound2 in <simp-all add: interval-lebesgue-integrable-def one-ereal-def set-borel-measurable-def f-def>*)  
**thus** *?thesis* **by** (*simp add: interval-lebesgue-integrable-def one-ereal-def*)  
**qed**$*

**lemma** *uniform-limit2:*

*uniform-limit {z. Re z ∈ {x..y}}*  
 $(\lambda a \ z. \text{LBINT } u = \exp(-1)..a. f \ z \ u)$   
 $(\lambda z. \text{LBINT } u = \exp(-1)..1. f \ z \ u)$  (*at-left 1*)  
**by** (*intro uniform-limit-interval-integral-right*[*of - - g2*] *AE-I2 impI*)  
*(use bound2 integrable-bound2 in <simp-all add: einterval-def f-def set-borel-measurable-def>)*

**lemma** *uniform-limit2':*

*uniform-limit {z. Re z ∈ {x..y}}*  
 $(\lambda n \ z. \text{LBINT } u = \exp(-1)..ereal (1 - (1/2)^{\wedge} \text{Suc } n). f \ z \ u)$   
 $(\lambda z. \text{LBINT } u = \exp(-1)..1. f \ z \ u)$  *sequentially*  
**proof** (*rule filterlim-compose*[*OF uniform-limit2*])  
**have** *filterlim*  $(\lambda n. 1 - (1/2)^{\wedge} \text{Suc } n :: \text{real})$  (*at-left 1*) *sequentially*  
**by** *real-asymp*

**hence**  $\text{filtermap } \text{ereal } (\text{filtermap } (\lambda n. (1 - (1 / 2) ^ \wedge \text{Suc } n)) \text{ sequentially}) \leq$   
 $\text{filtermap } \text{ereal } (\text{at-left } 1)$   
**unfolding**  $\text{filterlim-def}$  **by**  $(\text{rule } \text{filtermap-mono})$   
**thus**  $\text{filterlim } (\lambda n. \text{ereal } (1 - (1/2) ^ \wedge \text{Suc } n)) (\text{at-left } 1) \text{ sequentially}$   
**unfolding**  $\text{one-ereal-def at-left-ereal}$  **by**  $(\text{simp add: filterlim-def filtermap-filtermap})$   
**qed**

**lemma**  $\text{bound1: norm } (f z u) \leq g1 u$  **if**  $z: \text{Re } z \in \{x..y\}$  **and**  $u: u \in \{0 <.. < \text{exp } (-1)\}$  **for**  $z u$

**proof** –

**from**  $u$  **have**  $u \leq \text{exp } (-1)$  **by**  $(\text{simp add: einterval-def})$   
**also have**  $\text{exp } (-1) < \text{exp } (0::\text{real})$   
**by**  $(\text{subst exp-less-cancel-iff}) \text{ auto}$   
**also have**  $\text{exp } (0::\text{real}) = 1$  **by**  $\text{simp}$   
**finally have**  $u < 1$  .  
**from**  $u$  **have**  $\ln u < \ln (\text{exp } (-1))$   
**by**  $(\text{subst ln-less-cancel-iff}) (\text{auto simp: einterval-def})$   
**hence**  $\ln-u: \ln u < -1$  **by**  $\text{simp}$   
**from**  $z u \langle u < 1 \rangle$  **have**  $\text{norm } (f z u) = (- \ln u) \text{ powr } \text{Re } z * |u - \ln u - 1| /$   
 $(1 - u)$   
**unfolding**  $f\text{-def norm-mult norm-divide norm-of-real}$   
**by**  $(\text{simp add: norm-powr-real-powr einterval-def})$   
**also have**  $|u - \ln u - 1| = u - \ln u - 1$   
**using**  $u \text{ ln-add-one-self-le-self2[of } u - 1]$  **by**  $(\text{simp add: einterval-def})$   
**also have**  $(- \ln u) \text{ powr } \text{Re } z * (u - \ln u - 1) / (1 - u) \leq$   
 $(- \ln u) \text{ powr } y * (u - \ln u - 1) / (1 - u)$   
**using**  $z u \text{ ln-u } \langle u < 1 \rangle$   
**by**  $(\text{intro mult-right-mono divide-right-mono powr-mono}) (\text{auto simp: einterval-def})$   
**finally show**  $\text{norm } (f z u) \leq g1 u$  **by**  $(\text{simp add: g1-def})$   
**qed**

**lemma**  $\text{integrable1-aux: interval-lebesgue-integrable lborel } 0 (\text{exp } (-1)) (f z)$   
**if**  $z: \text{Re } z \in \{x..y\}$  **for**  $z$

**proof** –

**have**  $\text{set-integrable lborel } \{0 <.. < \text{exp } (-1)\} (f z)$   
**proof**  $(\text{rule set-integrable-bound[OF - - AE-I2[OF impI]])}$   
**fix**  $u :: \text{real}$  **assume**  $u \in \{0 <.. < \text{exp } (-1)\}$   
**hence**  $\text{norm } (f z u) \leq g1 u$  **using**  $z$  **by**  $(\text{intro bound1}) \text{ auto}$   
**also have**  $\dots \leq \text{norm } (g1 u)$  **by**  $\text{simp}$   
**finally show**  $\text{norm } (f z u) \leq \text{norm } (g1 u)$  .  
**qed**  $(\text{use integrable-bound1 in } \langle \text{simp-all add: interval-lebesgue-integrable-def zero-ereal-def set-borel-measurable-def f-def} \rangle)$   
**thus**  $?thesis$  **by**  $(\text{simp add: interval-lebesgue-integrable-def zero-ereal-def})$   
**qed**

**lemma**  $\text{uniform-limit1:}$

$\text{uniform-limit } \{z. \text{Re } z \in \{x..y\}\}$   
 $(\lambda a z. \text{LBINT } u=a.. \text{exp } (-1). f z u)$



$(\lambda z. \text{LBINT } u=0..exp (-1). f z u) \text{ (at-right 0)}$   
**by** (intro uniform-limit-interval-integral-left[of - - g1] AE-I2 impI)  
 (use bound1 integrable-bound1 in ⟨simp-all add: einterval-def f-def set-borel-measurable-def⟩)

**lemma** uniform-limit1':  
 uniform-limit {z. Re z ∈ {x..y}}  
 $(\lambda n z. \text{LBINT } u=ereal ((1/2)^{\wedge} \text{Suc } n)..exp (-1). f z u)$   
 $(\lambda z. \text{LBINT } u=0..exp (-1). f z u)$  sequentially  
**proof** (rule filterlim-compose[OF uniform-limit1])  
**have** filterlim  $(\lambda n. (1/2)^{\wedge} \text{Suc } n :: \text{real})$  (at-right 0) sequentially  
**by** real-asymp  
**hence** filtermap ereal (filtermap  $(\lambda n. ((1 / 2)^{\wedge} \text{Suc } n))$  sequentially) ≤  
 filtermap ereal (at-right 0)  
**unfolding** filterlim-def **by** (rule filtermap-mono)  
**thus** filterlim  $(\lambda n. \text{ereal } ((1/2)^{\wedge} \text{Suc } n))$  (at-right 0) sequentially  
**unfolding** zero-ereal-def at-right-ereal **by** (simp add: filterlim-def filtermap-filtermap)  
**qed**

**end**

With all of the above bounds, we have shown that the integral exists for any  $z$  with  $\Re(z) > -2$ .

**theorem** Hadjicostas-integral-integrable: interval-lebesgue-integrable lborel 0 1 (f z)  
**if** z: Re z > -2  
**proof** –  
**from** dense[OF z] **obtain** x **where** x: x > -2 Re z > x **by** blast  
**have** interval-lebesgue-integrable lborel 0 (exp(-1)) (f z)  
**by** (rule integrable1-aux[of x - Re z + 1]) (use x in auto)  
**moreover** **have** interval-lebesgue-integrable lborel (exp(-1)) 1 (f z)  
**by** (rule integrable2-aux[of x - Re z + 1]) (use x in auto)  
**ultimately show** interval-lebesgue-integrable lborel 0 1 (f z)  
**by** (rule interval-lebesgue-integrable-combine) (auto simp: f-def set-borel-measurable-def)  
**qed**

**lemma** integral-holo-aux:  
**assumes** ab: a > 0 a ≤ b b < 1  
**shows**  $(\lambda z. \text{LBINT } u=ereal a..ereal b. f z u)$  holomorphic-on A  
**proof** –  
**define** f' :: complex ⇒ real ⇒ complex  
**where** f' ≡  $(\lambda z u. \ln (-\ln u) * f z u)$   
**note** [derivative-intros] = has-field-derivative-complex-powr-right'  
  
**have**  $(\lambda z. \text{integral } (cbox a b) (f z))$  holomorphic-on UNIV  
**proof** (rule leibniz-rule-holomorphic[of - - - f'], goal-cases)  
**case** (1 z t)  
**show** ?case **unfolding** f-def  
**apply** (insert 1 ab)  
**apply** (rule derivative-eq-intros refl | simp)+

```

    apply (auto simp: f'-def field-simps f-def Ln-of-real)
  done
next
from ab show continuous-on (UNIV × cbox a b) (λ(z, t). f' z t)
  by (auto simp: case-prod-unfold f'-def f-def Ln-of-real intro!: continuous-intros)
next
fix z :: complex
show f z integrable-on cbox a b
  unfolding f-def f'-def using ab
  by (intro integrable-continuous continuous-intros) auto
qed (auto simp: convex-halfspace-Re-gt)
also have (λz. integral (cbox a b) (f z)) = (λz. ∫ u∈cbox a b. f z u ∂lborel)
proof (intro ext set-borel-integral-eq-integral(2) [symmetric])
  fix z :: complex
  show complex-set-integrable lborel (cbox a b) (f z)
    unfolding f-def using ab
    by (intro continuous-on-imp-set-integrable-cbox continuous-intros) (auto simp:
Ln-of-real)
  qed
  also have ... = (λz. LBINT u=a..b. f z u)
    using ab by (simp add: interval-integral-Icc)
  finally show ?thesis by (rule holomorphic-on-subset) auto
qed

```

**lemma** *integral-holo*:

```

  assumes ab: min a b > 0 max a b < 1
  shows (λz. LBINT u=ereal a..ereal b. f z u) holomorphic-on A
proof (cases a ≤ b)
  case True
  thus ?thesis using assms integral-holo-aux[of a b] by auto
next
  case False
  have (λz. -(LBINT u=ereal b..ereal a. f z u)) holomorphic-on A
    using False assms by (intro holomorphic-intros integral-holo-aux) auto
  thus ?thesis by (subst interval-integral-endpoints-reverse)
qed

```

**lemma** *holo1*: (λz. LBINT u=0..exp (-1). f z u) holomorphic-on {z. Re z > -2}

```

proof (rule holomorphic-uniform-sequence
  [where f = (λn z. LBINT u=ereal ((1/2) ^ Suc n)..exp (-1). f z u)], goal-cases)
  case (3 z)
  define ε where ε = (Re z + 2) / 2
  from 3 have ε > 0 by (auto simp: ε-def)
  have subset: cball z ε ⊆ {s. Re s ∈ {Re z - ε..Re z + ε}}
  proof safe
    fix s assume s: s ∈ cball z ε
    have |Re (s - z)| ≤ norm (s - z) by (rule abs-Re-le-cmod)
    also have ... ≤ ε using s by (simp add: dist-norm norm-minus-commute)
    finally show Re s ∈ {Re z - ε..Re z + ε} by auto
  qed

```

```

qed

show ?case
proof (rule exI[of - ε], intro conjI)
  have cball z ε ⊆ {s. Re s ∈ {Re z - ε..Re z + ε}} by fact
  also have ... ⊆ {s. Re s > -2}
  using 3 by (auto simp: ε-def field-simps)
  finally show cball z ε ⊆ {s. Re s > -2} .
next
  from 3 have Re z - ε > -2 by (simp add: ε-def field-simps)
  thus uniform-limit (cball z ε) (λn z. LBINT u=ereal ((1 / 2) ^ Suc n)..ereal
(exp (- 1)). f z u)
    (λz. LBINT u=0..ereal (exp(-1)). f z u) sequentially
    using uniform-limit-on-subset[OF uniform-limit1' subset] by simp
qed fact+
next
  fix n :: nat
  have (1 / 2) ^ Suc n < (1 / 2 :: real) ^ 0
    by (subst power-strict-decreasing-iff) auto
  thus (λz. LBINT u=ereal ((1 / 2) ^ Suc n)..ereal (exp (- 1)). f z u) holomor-
phic-on {z. Re z > -2}
    by (intro integral-holo) auto
qed (auto simp: open-halfspace-Re-gt)

lemma holo2: (λz. LBINT u=exp (-1)..1. f z u) holomorphic-on {z. Re z > -2}
proof (rule holomorphic-uniform-sequence
  [where f = (λn z. LBINT u=exp (-1)..ereal (1-(1/2)^Suc n). f z u)],
goal-cases)
  case (3 z)
  define ε where ε = (Re z + 2) / 2
  from 3 have ε > 0 by (auto simp: ε-def)
  have subset: cball z ε ⊆ {s. Re s ∈ {Re z - ε..Re z + ε}}
  proof safe
    fix s assume s: s ∈ cball z ε
    have |Re (s - z)| ≤ norm (s - z) by (rule abs-Re-le-cmod)
    also have ... ≤ ε using s by (simp add: dist-norm norm-minus-commute)
    finally show Re s ∈ {Re z - ε..Re z + ε} by auto
  qed
qed

show ?case
proof (rule exI[of - ε], intro conjI)
  have cball z ε ⊆ {s. Re s ∈ {Re z - ε..Re z + ε}} by fact
  also have ... ⊆ {s. Re s > -2}
  using 3 by (auto simp: ε-def field-simps)
  finally show cball z ε ⊆ {s. Re s > -2} .
next
  from 3 have Re z - ε > -2 by (simp add: ε-def field-simps)
  thus uniform-limit (cball z ε) (λn z. LBINT u=ereal (exp (- 1))..ereal (1-(1/2)^Suc
n). f z u)

```

```

      (λz. LBINT u=ereal (exp(-1))..1. f z u) sequentially
    using uniform-limit-on-subset[OF uniform-limit2' subset] by simp
  qed fact+
next
  fix n :: nat
  have (1 / 2) ^ Suc n < (1 / 2 :: real) ^ 0
    by (subst power-strict-decreasing-iff) auto
  thus (λz. LBINT u=ereal (exp (-1))..ereal (1-(1/2)^Suc n). f z u) holomor-
phic-on {z. Re z > -2}
    by (intro integral-holo) auto
  qed (auto simp: open-halfspace-Re-gt)

```

Finally, we have shown that Hadjicostas's integral is an analytic function of  $z$  in the domain  $\Re(z) > -2$ .

**lemma** *holomorphic-Hadjicostas-integral*:

*Hadjicostas-integral holomorphic-on {z. Re z > -2}*

**proof** –

**have** (λz. (LBINT u=0..exp(-1). f z u) + (LBINT u=exp(-1)..1. f z u))  
holomorphic-on {z. Re z > -2}

**by** (intro holomorphic-intros holo1 holo2)

**also have** ?this  $\longleftrightarrow$  (λz. LBINT u=0..1. f z u) holomorphic-on {z. Re z > -2}

**using** *Hadjicostas-integral-integrable*

**by** (intro holomorphic-cong interval-integral-sum)

(simp-all add: zero-ereal-def one-ereal-def min-def max-def)

**also have** (λz. LBINT u=0..1. f z u) = *Hadjicostas-integral*

**by** (simp add: *Hadjicostas-integral-def[abs-def] f-def*)

**finally show** ?thesis .

**qed**

**lemma** *analytic-Hadjicostas-integral*:

*Hadjicostas-integral analytic-on {z. Re z > -2}*

**by** (simp add: *analytic-on-open open-halfspace-Re-gt holomorphic-Hadjicostas-integral*)

**end**

### 4.3 Analytic continuation and main result

Since we have already shown the formula for any real  $z > -1$  and e. g. 0 is a limit point of that set, it extends to the full domain by analytic continuation.

As a caveat, note that  $\zeta(s)$  is *not* analytic at  $z = 1$ , so that we use an analytic continuation of  $\zeta(z) - \frac{1}{z-1}$  to state the formula. This continuation is *pre-zeta 1*.

**lemma** *Hadjicostas-Chapman-formula-aux*:

**assumes**  $z$ :  $\Re z > -2$

**shows** *Hadjicostas-integral*  $z = \text{Gamma } (z + 2) * \text{pre-zeta } 1 (z + 2)$

(is -  $z = ?f z$ )

**proof** (*rule analytic-continuation'[of Hadjicostas-integral]*)

**show** *Hadjicostas-integral holomorphic-on {z. Re z > -2}*

```

    by (rule holomorphic-Hadjicostas-integral)
  show connected {z. Re z > -2}
    by (intro convex-connected convex-halfspace-Re-gt)
  show open {z. Re z > -2}
    by (auto simp: open-halfspace-Re-gt)
  show {z. Re z > -1 ∧ Im z = 0} ⊆ {z. Re z > -2} and 0 ∈ {z. Re z > -2}
    by auto
  have ∀ n. 1 / (of-nat (Suc n)) ∈ {z. Re z > -1 ∧ Im z = 0} - {0}
    by (auto simp: field-simps simp flip: of-nat-Suc)
  moreover have (λ n. 1 / of-nat (Suc n) :: complex) ⟶ 0
    by (rule tendsto-divide-0[OF tendsto-const] filterlim-compose[OF tendsto-of-nat]
  filterlim-Suc)+
  ultimately show 0 islimpt {z. Re z > -1 ∧ Im z = 0}
    unfolding islimpt-sequential
    by (intro exI[of - λ n. 1 / of-nat (Suc n) :: complex]) simp
  show ?f holomorphic-on {z. -2 < Re z}
  proof (intro holomorphic-intros)
    fix z assume z: z ∈ {z. Re z > -2}
    hence z + 2 ∉ ℝ≤0 by (auto elim!: nonpos-Reals-cases simp: complex-eq-iff)
    thus z + 2 ∉ ℤ≤0 using nonpos-Ints-subset-nonpos-Reals by blast
  qed auto
next
  fix s assume s: s ∈ {z. -1 < Re z ∧ Im z = 0}
  hence s + 2 ≠ 1 by (simp add: algebra-simps complex-eq-iff)
  have ineq: x - ln x ≥ 1 if x ∈ {0 <..<1} for x :: real
    using ln-le-minus-one[of x] that by (simp add: algebra-simps)
  define x where x = Re s
  from s have x: x > -1 and [simp]: s = of-real x
    by (auto simp: x-def complex-eq-iff)
  have Hadjicostas-integral s = (LBINT u=0..1. of-real ((-ln u) powr x / (1-u)
  * (-ln u + u - 1)))
    unfolding Hadjicostas-integral-def
    by (intro interval-lebesgue-integral-cong) (auto simp: einterval-def powr-Reals-eq)
  also have ... = of-real (LBINT u=0..1. (-ln u) powr x / (1-u) * (-ln u + u
  - 1))
    by (subst interval-lebesgue-integral-of-real) auto
  also have (LBINT u=0..1. (-ln u) powr x / (1-u) * (-ln u + u - 1)) =
    (∫ u. (-ln u) powr x / (1-u) * (-ln u + u - 1) * indicator {0 <..<1}
  u ∂lborel)
    by (simp add: interval-integral-Ioo zero-ereal-def one-ereal-def set-lebesgue-integral-def
  mult-ac)
  also have ... = enn2real (Hadjicostas-nn-integral x)
    unfolding Hadjicostas-nn-integral-def using ineq
    by (subst integral-eq-nn-integral)
    (auto intro!: AE-I2 divide-nonneg-nonneg mult-nonneg-nonneg arg-cong[where
  f = enn2real])
    nn-integral-cong simp: indicator-def)
  also have ... = enn2real (ennreal (Gamma (x + 2) * (Re (zeta (x + 2)) - 1
  / (x + 1))))

```

**using**  $x$  **by** (*subst Hadjicostas-Chapman-formula-real*) *auto*  
**also have**  $\dots = \text{Gamma}(x + 2) * (\text{Re}(\text{zeta}(x + 2)) - 1 / (x + 1))$   
**using**  $x$  *real-zeta-ge-one-over-minus-one*[*of*  $x + 2$ ]  
**by** (*intro enn2real-ennreal mult-nonneg-nonneg Gamma-real-nonneg*) (*auto simp: add-ac*)  
**also have** *complex-of-real*  $\dots = \text{Gamma}(s + 2) * (\text{zeta}(s + 2) - 1 / (s + 1))$   
**using**  $x$  *Gamma-complex-of-real*[*of*  $x + 2$ ] **by** (*simp add: zeta-real*)  
**also have**  $(\text{zeta}(s + 2) - 1 / (s + 1)) = \text{pre-zeta } 1 (s + 2)$   
**using**  $\langle s + 2 \neq 1 \rangle$  **by** (*subst zeta-minus-pole-eq [symmetric]*) (*auto simp flip: of-nat-Suc*)  
**finally show** *Hadjicostas-integral*  $s = \text{Gamma}(s + 2) * \text{pre-zeta } 1 (s + 2)$  .  
**qed** (*use assms in auto*)

The following form and the corollary are perhaps a bit nicer to read.

**theorem** *Hadjicostas-Chapman-formula*:  
**assumes**  $z: \text{Re } z > -2 \ z \neq -1$   
**shows** *Hadjicostas-integral*  $z = \text{Gamma}(z + 2) * (\text{zeta}(z + 2) - 1 / (z + 1))$   
**proof** –  
**from**  $z$  **have**  $z + 1 \neq 0$   
**by** (*auto simp: complex-eq-iff*)  
**thus** *?thesis* **using** *Hadjicostas-Chapman-formula-aux*[*of*  $z$ ] *assms*  
**by** (*subst (asm) zeta-minus-pole-eq [symmetric]*) (*auto simp: add-ac*)  
**qed**

**corollary** *euler-mascheroni-integral-form*:  
*Hadjicostas-integral*  $(-1) = \text{euler-mascheroni}$   
**using** *Hadjicostas-Chapman-formula-aux*[*of*  $-1$ ] **by** *simp*

**end**

## References

- [1] T. M. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer-Verlag, 1976.
- [2] W. E. Briggs and S. Chowla. The power series coefficients of  $\zeta(s)$ . *The American Mathematical Monthly*, 62(5):323–325, 1955.
- [3] R. Chapman. A proof of Hadjicostas’s conjecture, 2004.
- [4] P. Hadjicostas. A conjecture-generalization of Sondow’s formula, 2004.
- [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.
- [7] J. Sondow. Double integrals for Euler's constant and  $\ln \frac{4}{\pi}$ , 2002.