

# Irrationality Criteria for Series by Erdős and Straus

Angeliki Koutsoukou-Argyraki and Wenda Li

June 6, 2026

## Abstract

We formalise certain irrationality criteria for infinite series of the form:

$$\sum_n \frac{b_n}{\prod_{i \leq n} a_i}$$

where  $b_n, a_i$  are integers. The result is due to P. Erdős and E.G. Straus [1], and in particular we formalise Theorem 2.1, Corollary 2.10 and Theorem 3.1. The latter is an application of Theorem 2.1 involving the prime numbers.

## Contents

<b>1</b>	<b>Miscellaneous</b>	<b>1</b>
<b>2</b>	<b>Theorem 2.1 and Corollary 2.10</b>	<b>3</b>
<b>3</b>	<b>Some auxiliary results on the prime numbers.</b>	<b>21</b>
<b>4</b>	<b>Theorem 3.1</b>	<b>23</b>
<b>5</b>	<b>Acknowledgements</b>	<b>42</b>

```
theory Irrational-Series-Erdos-Straus imports  
  Prime-Number-Theorem.Prime-Number-Theorem  
  Prime-Distribution-Elementary.PNT-Consequences  
begin
```

## 1 Miscellaneous

```
lemma suminf-comparison:  
  assumes summable f and gf:  $\bigwedge n. \text{norm } (g\ n) \leq f\ n$   
  shows suminf g  $\leq$  suminf f  
proof (rule suminf-le)  
  show  $g\ n \leq f\ n$  for  $n$ 
```

```

    using gf[of n] by auto
  show summable g
    using assms summable-comparison-test' by blast
  show summable f using assms(1) .
qed

```

```

lemma tendsto-of-int-diff-0:
  assumes (λn. f n - of-int(g n)) ⟶ (0::real) ∀F n in sequentially. f n > 0
  shows ∀F n in sequentially. 0 ≤ g n
proof -
  have ∀F n in sequentially. |f n - of-int(g n)| < 1 / 2
    using assms(1)[unfolded tendsto-iff,rule-format,of 1/2] by auto
  then show ?thesis using assms(2)
    by eventually-elim linarith
qed

```

```

lemma eventually-mono-sequentially:
  assumes eventually P sequentially
  assumes ∧x. P (x+k) ⟹ Q (x+k)
  shows eventually Q sequentially
  using sequentially-offset[OF assms(1),of k]
  apply (subst eventually-sequentially-seg[symmetric,of - k])
  apply (elim eventually-mono)
  by fact

```

```

lemma frequently-eventually-at-top:
  fixes P Q::'a::linorder ⇒ bool
  assumes frequently P at-top eventually Q at-top
  shows frequently (λx. P x ∧ (∀y≥x. Q y) ) at-top
  using assms
  unfolding frequently-def eventually-at-top-linorder
  by (metis (mono-tags, opaque-lifting) le-cases order-trans)

```

```

lemma eventually-at-top-mono:
  fixes P Q::'a::linorder ⇒ bool
  assumes event-P:eventually P at-top
  assumes PQ-imp:∧x. x≥z ⟹ ∀y≥x. P y ⟹ Q x
  shows eventually Q at-top
proof -
  obtain N where ∀n≥N. P n
    by (meson event-P eventually-at-top-linorder)
  then have Q x when x ≥ max N z for x
    using PQ-imp that by auto
  then show ?thesis unfolding eventually-at-top-linorder
    by blast
qed

```

```

lemma frequently-at-top-elim:
  fixes P Q::'a::linorder ⇒ bool

```

**assumes**  $\exists_F x$  in at-top.  $P x$   
**assumes**  $\bigwedge i. P i \implies \exists j > i. Q j$   
**shows**  $\exists_F x$  in at-top.  $Q x$   
**using** *assms unfolding frequently-def eventually-at-top-linorder*  
**by** (*meson leD le-cases less-le-trans*)

**lemma** *less-Liminf-iff*:  
**fixes**  $X :: - \implies - :: \text{complete-linorder}$   
**shows**  $\text{Liminf } F X < C \iff (\exists y < C. \text{frequently } (\lambda x. y \geq X x) F)$   
**by** (*force simp: not-less not-frequently not-le le-Liminf-iff simp flip: Not-eq-iff*)

**lemma** *sequentially-even-odd-imp*:  
**assumes**  $\forall_F N$  in sequentially.  $P (2*N) \forall_F N$  in sequentially.  $P (2*N+1)$   
**shows**  $\forall_F n$  in sequentially.  $P n$

**proof** –  
**obtain**  $N$  **where**  $N-P: \forall x \geq N. P (2 * x) \wedge P (2 * x + 1)$   
**using** *eventually-conj[OF assms]*  
**unfolding** *eventually-at-top-linorder* **by** *auto*  
**have**  $P n$  **when**  $n \geq 2*N$  **for**  $n$   
**proof** –  
**define**  $n'$  **where**  $n' = n \text{ div } 2$   
**then have**  $n' \geq N$  **using** *that* **by** *auto*  
**then have**  $P (2 * n') \wedge P (2 * n' + 1)$   
**using**  $N-P$  **by** *auto*  
**then show** *?thesis* **unfolding**  $n'$ -*def*  
**by** (*cases even n*) *auto*  
**qed**  
**then show** *?thesis* **unfolding** *eventually-at-top-linorder* **by** *auto*  
**qed**

## 2 Theorem 2.1 and Corollary 2.10

**context**  
**fixes**  $a b :: \text{nat} \implies \text{int}$   
**assumes**  $a\text{-pos}: \forall n. a n > 0$  **and**  $a\text{-large}: \forall_F n$  in sequentially.  $a n > 1$   
**and**  $ab\text{-tendsto}: (\lambda n. |b n| / (a (n-1) * a n)) \longrightarrow 0$   
**begin**

**private lemma** *aux-series-summable*: *summable*  $(\lambda n. b n / (\prod_{k \leq n} a k))$   
**proof** –  
**have**  $\bigwedge e. e > 0 \implies \forall_F x$  in sequentially.  $|b x| / (a (x-1) * a x) < e$   
**using** *ab-tendsto[unfolded tendsto-iff]*  
**apply** (*simp add: abs-mult flip: of-int-abs*)  
**by** (*subst (asm) (2) abs-of-pos, use  $\langle \forall n. a n > 0 \rangle$  in auto*) +  
**from** *this[of 1]*  
**have**  $\forall_F x$  in sequentially.  $|\text{real-of-int}(b x)| < (a (x-1) * a x)$   
**using**  $\langle \forall n. a n > 0 \rangle$  **by** *auto*  
**moreover have**  $\forall n. (\prod_{k \leq n} \text{real-of-int}(a k)) > 0$   
**using**  $a\text{-pos}$  **by** (*auto intro!: linordered-semidom-class.prod-pos*)

**ultimately have**  $\forall_F n$  in sequentially.  $|b\ n| / (\prod_{k \leq n}. a\ k)$   
 $< (a\ (n-1) * a\ n) / (\prod_{k \leq n}. a\ k)$   
**apply** (*elim eventually-mono*)  
**by** (*auto simp:field-simps*)  
**moreover have**  $|b\ n| / (\prod_{k \leq n}. a\ k) = \text{norm } (b\ n / (\prod_{k \leq n}. a\ k))$  **for**  $n$   
**using**  $\langle \forall n. (\prod_{k \leq n}. \text{real-of-int } (a\ k)) > 0 \rangle$  [*rule-format, of n*] **by** *auto*  
**ultimately have**  $\forall_F n$  in sequentially.  $\text{norm } (b\ n / (\prod_{k \leq n}. a\ k))$   
 $< (a\ (n-1) * a\ n) / (\prod_{k \leq n}. a\ k)$   
**by** *algebra*  
**moreover have** *summable*  $(\lambda n. (a\ (n-1) * a\ n) / (\prod_{k \leq n}. a\ k))$   
**proof** –  
**obtain**  $s$  **where** *a-gt-1*:  $\forall n \geq s. a\ n > 1$   
**using** *a-large* [*unfolded eventually-at-top-linorder*] **by** *auto*  
**define**  $cc$  **where**  $cc = (\prod_{k < s}. a\ k)$   
**have**  $cc > 0$   
**unfolding** *cc-def* **by** (*meson a-pos prod-pos*)  
**have**  $(\prod_{k \leq n+s}. a\ k) \geq cc * 2^n$  **for**  $n$   
**proof** –  
**have**  $\text{prod } a\ \{s..< \text{Suc } (s + n)\} \geq 2^n$   
**proof** (*induct n*)  
**case**  $0$   
**then show** *?case* **using** *a-gt-1* **by** *auto*  
**next**  
**case** (*Suc n*)  
**moreover have**  $a\ (s + \text{Suc } n) \geq 2$   
**by** (*smt (verit, ccfv-threshold) a-gt-1 le-add1*)  
**ultimately show** *?case*  
**apply** (*subst prod.atLeastLessThan-Suc,simp*)  
**using** *mult-mono'* [*of 2 a (Suc (s + n)) 2^n prod a {s..<Suc (s + n)}*]  
**by** (*simp add: mult commute*)  
**qed**  
**moreover have**  $\text{prod } a\ \{0..(n + s)\} = \text{prod } a\ \{..<s\} * \text{prod } a\ \{s..<\text{Suc } (s + n)\}$   
**using** *prod.atLeastLessThan-concat* [*of 0 s s+n+1 a*]  
**by** (*simp add: add commute lessThan-atLeast0 prod.atLeastLessThan-concat prod.head-if*)  
**ultimately show** *?thesis*  
**using**  $\langle cc > 0 \rangle$  **unfolding** *cc-def* **by** (*simp add: atLeast0AtMost*)  
**qed**  
**then have**  $1 / (\prod_{k \leq n+s}. a\ k) \leq 1 / (cc * 2^n)$  **for**  $n$   
**proof** –  
**assume** *asm*:  $\bigwedge n. cc * 2^n \leq \text{prod } a\ \{..n + s\}$   
**then have**  $\text{real-of-int } (cc * 2^n) \leq \text{prod } a\ \{..n + s\}$  **using** *of-int-le-iff* **by**  
*blast*  
**moreover have**  $\text{prod } a\ \{..n + s\} > 0$  **using**  $\langle cc > 0 \rangle$  **by** (*simp add: a-pos prod-pos*)  
**ultimately show** *?thesis* **using**  $\langle cc > 0 \rangle$   
**by** (*auto simp:field-simps simp del:of-int-prod*)  
**qed**

```

moreover have summable ( $\lambda n. 1 / (cc * 2^n)$ )
proof –
  have summable ( $\lambda n. 1 / (2::int)^n$ )
    using summable-geometric[of 1/(2::int)] by (simp add:power-one-over)
  from summable-mult[OF this, of 1/cc] show ?thesis by auto
qed
ultimately have summable ( $\lambda n. 1 / (\prod_{k \leq n+s} a k)$ )
  apply (elim summable-comparison-test'[where N=0])
  apply (unfold real-norm-def, subst abs-of-pos)
  by (auto simp:  $\langle \forall n. 0 < (\prod_{k \leq n} \text{real-of-int } (a k)) \rangle$ )
then have summable ( $\lambda n. 1 / (\prod_{k \leq n} a k)$ )
  apply (subst summable-iff-shift[where k=s,symmetric])
  by simp
then have summable ( $\lambda n. (a (n+1) * a (n+2)) / (\prod_{k \leq n+2} a k)$ )
proof –
  assume asm:summable ( $\lambda n. 1 / \text{real-of-int } (\text{prod } a \{..n\})$ )
  have  $1 / \text{real-of-int } (\text{prod } a \{..n\}) = (a (n+1) * a (n+2)) / (\prod_{k \leq n+2} a k)$ 
for  $n$ 
  proof –
    have  $a (Suc (Suc n)) \neq 0$   $a (Suc n) \neq 0$ 
      using a-pos by (metis less-irrefl)+
    then show ?thesis
      by (simp add: atLeast0-atMost-Suc atMost-atLeast0)
  qed
  then show ?thesis using asm by auto
qed
then show summable ( $\lambda n. (a (n-1) * a n) / (\prod_{k \leq n} a k)$ )
  apply (subst summable-iff-shift[symmetric, of - 2])
  by auto
qed
ultimately show ?thesis
  apply (elim summable-comparison-test-ev[rotated])
  by (simp add: eventually-mono)
qed

```

```

private fun get-c::(nat  $\Rightarrow$  int)  $\Rightarrow$  (nat  $\Rightarrow$  int)  $\Rightarrow$  int  $\Rightarrow$  nat  $\Rightarrow$  (nat  $\Rightarrow$  int) where
  get-c a' b' B N 0 = round (B * b' N / a' N)
  get-c a' b' B N (Suc n) = get-c a' b' B N n * a' (n+N) - B * b' (n+N)

```

**lemma** ab-rationality-imp:

**assumes** ab-rational:( $\sum n. (b n / (\prod_{i \leq n} a i)) \in \mathbb{Q}$ )

**shows**  $\exists (B::int) > 0. \exists c::nat \Rightarrow int.$

$(\forall_F n$  in sequentially.  $B * b n = c n * a n - c(n+1) \wedge |c(n+1)| < a n / 2$ )  
 $\wedge (\lambda n. c (Suc n) / a n) \longrightarrow 0$

**proof** –

**have** [simp]:  $a n \neq 0$  **for**  $n$  **using** a-pos **by** (metis less-numeral-extra(3))

**obtain**  $A::int$  **and**  $B::int$  **where**

$AB\text{-eq}:(\sum n. \text{real-of-int } (b n) / \text{real-of-int } (\text{prod } a \{..n\})) = A / B$  **and**  $B > 0$

**proof** –

```

obtain  $q::\text{rat}$  where  $(\sum n. \text{real-of-int } (b \ n) / \text{real-of-int } (\text{prod } a \ \{..n\})) =$ 
real-of-rat  $q$ 
using ab-rational by (rule Rats-cases) simp
moreover obtain  $A::\text{int}$  and  $B::\text{int}$  where  $q = \text{Rat.Fract } A \ B \ B > 0$  coprime
 $A \ B$ 
by (rule Rat-cases) auto
ultimately show ?thesis by (auto intro!: that[of  $A \ B$ ] simp:of-rat-rat)
qed
define  $f$  where  $f \equiv (\lambda n. b \ n / \text{real-of-int } (\text{prod } a \ \{..n\}))$ 
define  $R$  where  $R \equiv (\lambda N. (\sum n. B*b \ (n+N+1) / \text{prod } a \ \{N..n+N+1\}))$ 
have all-e-ubound: $\forall e>0. \forall_F M$  in sequentially.  $\forall n. |B*b \ (n+M+1) / \text{prod } a$ 
 $\{M..n+M+1\}| < e/4 * 1/2^{\wedge}n$ 
proof safe
fix  $e::\text{real}$  assume  $e>0$ 
obtain  $N$  where  $N\text{-a2}$ :  $\forall n \geq N. a \ n \geq 2$ 
and  $N\text{-ba}$ :  $\forall n \geq N. |b \ n| / (a \ (n-1) * a \ n) < e/(4*B)$ 
proof  $-$ 
have  $\forall_F n$  in sequentially.  $|b \ n| / (a \ (n-1) * a \ n) < e/(4*B)$ 
using order-topology-class.order-tendstoD[OF ab-tendsto,of  $e/(4*B)$ ]  $\langle B>0 \rangle$ 
 $\langle e>0 \rangle$ 
by auto
moreover have  $\forall_F n$  in sequentially.  $a \ n \geq 2$ 
using a-large by (auto elim: eventually-mono)
ultimately have  $\forall_F n$  in sequentially.  $|b \ n| / (a \ (n-1) * a \ n) < e/(4*B)$ 
 $\wedge a \ n \geq 2$ 
by eventually-elim auto
then show ?thesis unfolding eventually-at-top-linorder using that
by auto
qed
have geq-N-bound: $|B*b \ (n+M+1) / \text{prod } a \ \{M..n+M+1\}| < e/4 * 1/2^{\wedge}n$ 
when  $M \geq N$  for  $n \ M$ 
proof  $-$ 
define  $D$  where  $D = B*b \ (n+M+1) / (a \ (n+M) * a \ (n+M+1))$ 
have  $|B*b \ (n+M+1) / \text{prod } a \ \{M..n+M+1\}| = |D / \text{prod } a \ \{M..<n+M\}|$ 
proof  $-$ 
have  $\{M..n+M+1\} = \{M..<n+M\} \cup \{n+M, n+M+1\}$  by auto
then have  $\text{prod } a \ \{M..n+M+1\} = a \ (n+M) * a \ (n+M+1) * \text{prod } a$ 
 $\{M..<n+M\}$  by simp
then show ?thesis unfolding  $D\text{-def}$  by (simp add:algebra-simps)
qed
also have  $\dots < |e/4 * (1/\text{prod } a \ \{M..<n+M\})|$ 
proof  $-$ 
have  $|D| < e/4$ 
unfolding  $D\text{-def}$  using  $N\text{-ba}$ [rule-format, of  $n+M+1$ ]  $\langle B>0 \rangle \langle M \geq N \rangle$ 
 $\langle e>0 \rangle$  a-pos
by (auto simp:field-simps abs-mult abs-of-pos)
from mult-strict-right-mono[OF this,of  $1/\text{prod } a \ \{M..<n+M\}$ ] a-pos  $\langle e>0 \rangle$ 
show ?thesis
apply (auto simp:abs-prod abs-mult prod-pos)

```

```

    by (subst (2) abs-of-pos,auto)+
  qed
  also have ... ≤ e/4 * 1/2n
  proof -
    have prod a {M..n+M} ≥ 2n
    proof (induct n)
      case 0
      then show ?case by simp
    next
      case (Suc n)
      then show ?case
    using ⟨M≥N⟩ by (simp add: N-a2 mult.commute mult-mono' prod.atLeastLessThan-Suc)
  qed
  then have real-of-int (prod a {M..n+M}) ≥ 2n
    using numeral-power-le-of-int-cancel-iff by blast
  then show ?thesis using ⟨e>0⟩ by (auto simp:divide-simps)
  qed
  finally show ?thesis .
  qed
  show ∀F M in sequentially. ∀n. |real-of-int (B * b (n + M + 1))
    / real-of-int (prod a {M..n+M+1})| < e / 4 * 1 / 2n
  apply (rule eventually-sequentiallyI[of N])
  using geq-N-bound by blast
  qed
  have R-tendsto-0:R → 0
  proof (rule tendstoI)
    fix e::real assume e>0
    show ∀F x in sequentially. dist (R x) 0 < e using all-e-ubound[rule-format,OF
    ⟨e>0⟩]
  proof eventually-elim
    case (elim M)
    define g where g = (λn. B*b (n+M+1) / prod a {M..n+M+1})
    have g-lt:|g n| < e/4 * 1/2n for n
      using elim unfolding g-def by auto
    have §: summable (λn. (e/4) * (1/2)n)
      by simp
    then have g-abs-summable:summable (λn. |g n|)
      apply (elim summable-comparison-test')
      by (metis abs-idempotent g-lt less-eq-real-def power-one-over real-norm-def
    times-divide-eq-right)
    have |∑ n. g n| ≤ (∑ n. |g n|) by (rule summable-rabs[OF g-abs-summable])
    also have ... ≤ (∑ n. e/4 * 1/2n)
    proof (rule suminf-comparison)
      show summable (λn. e/4 * 1/2n)
        using § unfolding power-divide by simp
      show ∧n. norm |g n| ≤ e / 4 * 1 / 2n using g-lt less-eq-real-def by
    auto
  qed
  also have ... = (e/4) * (∑ n. (1/2)n)

```

**apply** (*subst suminf-mult[symmetric]*)  
**by** (*auto simp: algebra-simps power-divide*)  
**also have** ... =  $e/2$  **by** (*simp add:suminf-geometric[of 1/2]*)  
**finally have**  $|\sum n. g n| \leq e / 2$  .  
**then show**  $\text{dist } (R M) 0 < e$  **unfolding** *R-def g-def* **using**  $\langle e > 0 \rangle$  **by** *auto*  
**qed**  
**qed**

**obtain** *N* **where** *R-N-bound*:  $\forall M \geq N. |R M| \leq 1 / 4$   
**and** *N-geometric*:  $\forall M \geq N. \forall n. |\text{real-of-int } (B * b (n + M + 1)) / (\text{prod } a \{M..n + M + 1\})| < 1 / 2 ^ n$   
**proof** –  
**obtain** *N1* **where** *N1*:  $\forall M \geq N1. |R M| \leq 1 / 4$   
**using** *metric-LIMSEQ-D[OF R-tendsto-0, of 1/4]* *all-e-ubound[rule-format, of 4, unfolded eventually-sequentially]*  
**by** (*auto simp: less-eq-real-def*)  
**obtain** *N2* **where** *N2*:  $\forall M \geq N2. \forall n. |\text{real-of-int } (B * b (n + M + 1)) / (\text{prod } a \{M..n + M + 1\})| < 1 / 2 ^ n$   
**using** *all-e-ubound[rule-format, of 4, unfolded eventually-sequentially]*  
**by** (*auto simp: less-eq-real-def*)  
**define** *N* **where**  $N = \max N1 N2$   
**show** *?thesis* **using** *that[of N]* *N1 N2* **unfolding** *N-def* **by** *simp*  
**qed**

**define** *C* **where**  $C = B * \text{prod } a \{..<N\} * (\sum n < N. f n)$   
**have** *summable f*  
**unfolding** *f-def* **using** *aux-series-summable* .  
**have**  $A * \text{prod } a \{..<N\} = C + B * b N / a N + R N$   
**proof** –  
**have**  $A * \text{prod } a \{..<N\} = B * \text{prod } a \{..<N\} * (\sum n. f n)$   
**unfolding** *AB-eq f-def* **using**  $\langle B > 0 \rangle$  **by** *auto*  
**also have** ... =  $B * \text{prod } a \{..<N\} * ((\sum n < N+1. f n) + (\sum n. f (n+N+1)))$   
**using** *suminf-split-initial-segment[OF <summable f>, of N+1]* **by** *auto*  
**also have** ... =  $B * \text{prod } a \{..<N\} * ((\sum n < N. f n) + f N + (\sum n. f (n+N+1)))$   
**using** *sum.atLeast0-lessThan-Suc* **by** *simp*  
**also have** ... =  $C + B * b N / a N + (\sum n. B * b (n+N+1) / \text{prod } a \{N..n+N+1\})$   
**proof** –  
**have**  $B * \text{prod } a \{..<N\} * f N = B * b N / a N$   
**proof** –  
**have**  $\{..N\} = \{..<N\} \cup \{N\}$  **using** *ivl-disj-un-singleton(2)* **by** *blast*  
**then show** *?thesis* **unfolding** *f-def* **by** *auto*  
**qed**  
**moreover have**  $B * \text{prod } a \{..<N\} * (\sum n. f (n+N+1)) = (\sum n. B * b (n+N+1) / \text{prod } a \{N..n+N+1\})$   
**proof** –  
**have** *summable*  $(\lambda n. f (n + N + 1))$   
**using**  $\langle \text{summable } f \rangle$  *summable-iff-shift[of f N+1]* **by** *auto*  
**moreover have**  $\text{prod } a \{..<N\} * f (n + N + 1) = b (n + N + 1) / \text{prod}$

```

a {N..n + N + 1} for n
  proof -
    have {..n + N + 1} = {..<N} ∪ {N..n + N + 1} by auto
    then show ?thesis
      unfolding f-def
      apply simp
      apply (subst prod.union-disjoint)
      by auto
    qed
    ultimately show ?thesis
      apply (subst suminf-mult[symmetric])
      by (auto simp: mult commute mult.left-commute)
    qed
    ultimately show ?thesis unfolding C-def by (auto simp: algebra-simps)
    qed
    also have ... = C + B * b N / a N + R N
      unfolding R-def by simp
    finally show ?thesis .
  qed
  have R-bound: |R M| ≤ 1 / 4 and R-Suc: R (Suc M) = a M * R M - B * b
(Suc M) / a (Suc M)
  when M ≥ N for M
  proof -
    define g where g = (λn. B*b (n+M+1) / prod a {M..n+M+1})
    have g-abs-summable: summable (λn. |g n|)
    proof -
      have summable (λn. (1/2::real) ^ n)
        by simp
      moreover have |g n| < 1/2^n for n
        using N-geometric[rule-format, OF that] unfolding g-def by simp
      ultimately show ?thesis
        apply (elim summable-comparison-test')
        by (simp add: less-eq-real-def power-one-over)
    qed
    show |R M| ≤ 1 / 4 using R-N-bound[rule-format, OF that] .
    have R M = (∑ n. g n) unfolding R-def g-def by simp
    also have ... = g 0 + (∑ n. g (Suc n))
      apply (subst suminf-split-head)
      using summable-rabs-cancel[OF g-abs-summable] by auto
    also have ... = g 0 + 1/a M * (∑ n. a M * g (Suc n))
      apply (subst suminf-mult)
      by (auto simp: g-abs-summable summable-Suc-iff summable-rabs-cancel)
    also have ... = g 0 + 1/a M * R (Suc M)
    proof -
      have a M * g (Suc n) = B * b (n + M + 2) / prod a {Suc M..n + M + 2}
    for n
    proof -
      have {M..Suc (Suc (M + n))} = {M} ∪ {Suc M..Suc (Suc (M + n))} by
auto

```

**then show** *?thesis*  
**unfolding** *g-def* **using**  $\langle B > 0 \rangle$  **by** (*auto simp: algebra-simps*)  
**qed**  
**then have**  $(\sum n. a M * g (Suc n)) = R (Suc M)$   
**unfolding** *R-def* **by** *auto*  
**then show** *?thesis* **by** *auto*  
**qed**  
**finally have**  $R M = g 0 + 1 / a M * R (Suc M)$  .  
**then have**  $R (Suc M) = a M * R M - g 0 * a M$   
**by** (*auto simp: algebra-simps*)  
**moreover have**  $\{M..Suc M\} = \{M, Suc M\}$  **by** *auto*  
**ultimately show**  $R (Suc M) = a M * R M - B * b (Suc M) / a (Suc M)$   
**unfolding** *g-def* **by** *auto*  
**qed**

**define** *c* **where**  $c = (\lambda n. \text{if } n \geq N \text{ then } get-c \ a \ b \ B \ N \ (n - N) \ \text{else } \text{undefined})$   
**have** *c-rec*:  $c (n + 1) = c n * a n - B * b n$  **when**  $n \geq N$  **for** *n*  
**unfolding** *c-def* **using** *that* **by** (*auto simp: Suc-diff-le*)  
**have** *c-R*:  $c (Suc n) / a n = R n$  **when**  $n \geq N$  **for** *n*  
**using** *that*  
**proof** (*induct rule: nat-induct-at-least*)  
**case** *base*  
**have**  $|c (N + 1) / a N| \leq 1/2$   
**proof** –  
**have**  $c N = \text{round } (B * b N / a N)$  **unfolding** *c-def* **by** *simp*  
**moreover have**  $c (N + 1) / a N = c N - B * b N / a N$   
**using** *a-pos[rule-format, of N]*  
**by** (*auto simp: c-rec[of N, simplified] divide-simps*)  
**ultimately show** *?thesis* **using** *of-int-round-abs-le* **by** *auto*  
**qed**  
**moreover have**  $|R N| \leq 1 / 4$  **using** *R-bound[of N]* **by** *simp*  
**ultimately have**  $|c (N + 1) / a N - R N| < 1$  **by** *linarith*  
**moreover have**  $c (N + 1) / a N - R N \in \mathbb{Z}$   
**proof** –  
**have**  $c (N + 1) / a N = c N - B * b N / a N$   
**using** *a-pos[rule-format, of N]*  
**by** (*auto simp: c-rec[of N, simplified] divide-simps*)  
**moreover have**  $B * b N / a N + R N \in \mathbb{Z}$   
**proof** –  
**have**  $C = B * (\sum n < N. \text{prod } a \ \{..<N\} * (b n / \text{prod } a \ \{..n\}))$   
**unfolding** *C-def f-def* **by** (*simp add: sum-distrib-left algebra-simps*)  
**also have**  $\dots = B * (\sum n < N. \text{prod } a \ \{n<..<N\} * b n)$   
**proof** –  
**have**  $\{..<N\} = \{n<..<N\} \cup \{..n\}$  **if**  $n < N$  **for** *n*  
**by** (*simp add: ivl-disj-un-one(1) sup-commute that*)  
**then show** *?thesis*  
**using**  $\langle B > 0 \rangle$   
**apply** *simp*  
**apply** (*subst prod.union-disjoint*)

```

    by auto
  qed
  finally have  $C = \text{real-of-int } (B * (\sum n < N. \text{prod } a \{n < .. < N\} * b \ n))$  .
  then have  $C \in \mathbb{Z}$  using Ints-of-int by blast
  moreover note  $\langle A * \text{prod } a \{.. < N\} = C + B * b \ N / a \ N + R \ N \rangle$ 
  ultimately show ?thesis
    by (metis Ints-diff Ints-of-int add.assoc add-diff-cancel-left')
  qed
  ultimately show ?thesis by (simp add: diff-diff-add)
  qed
  ultimately have  $c \ (N+1) / a \ N - R \ N = 0$ 
    by (metis Ints-cases less-irrefl of-int-0 of-int-lessD)
  then show ?case by simp
next
  case (Suc n)
  have  $c \ (\text{Suc } (\text{Suc } n)) / a \ (\text{Suc } n) = c \ (\text{Suc } n) - B * b \ (\text{Suc } n) / a \ (\text{Suc } n)$ 
    apply (subst c-rec[of Suc n,simplified])
    using  $\langle N \leq n \rangle$  by (auto simp: divide-simps)
  also have  $\dots = a \ n * R \ n - B * b \ (\text{Suc } n) / a \ (\text{Suc } n)$ 
    using Suc by (auto simp: divide-simps)
  also have  $\dots = R \ (\text{Suc } n)$ 
    using R-Suc[OF <N ≤ n>] by simp
  finally show ?case .
  qed
  have ca-tendsto-zero:  $(\lambda n. c \ (\text{Suc } n) / a \ n) \longrightarrow 0$ 
    using R-tendsto-0
    apply (elim filterlim-mono-eventually)
    using c-R by (auto intro!: eventually-sequentiallyI[of N])
  have ca-bound:  $|c \ (n + 1)| < a \ n / 2$  when  $n \geq N$  for  $n$ 
  proof -
    have  $|c \ (\text{Suc } n)| / a \ n = |c \ (\text{Suc } n) / a \ n|$  using a-pos[rule-format,of n] by
  auto
    also have  $\dots = |R \ n|$  using c-R[OF that] by auto
    also have  $\dots < 1/2$  using R-bound[OF that] by auto
    finally have  $|c \ (\text{Suc } n)| / a \ n < 1/2$  .
    then show ?thesis using a-pos[rule-format,of n] by auto
  qed
  show  $\exists B > 0. \exists c. (\forall_F n \text{ in sequentially. } B * b \ n = c \ n * a \ n - c \ (n + 1))$ 
     $\wedge \text{real-of-int } |c \ (n + 1)| < a \ n / 2 \wedge (\lambda n. c \ (\text{Suc } n) / a \ n) \longrightarrow 0$ 
    unfolding eventually-at-top-linorder
    apply (rule exI[of - B], use  $\langle B > 0 \rangle$  in simp)
    apply (intro exI[of - c] exI[of - N])
    using c-rec ca-bound ca-tendsto-zero
    by fastforce
  qed
  private lemma imp-ab-rational:

```

**assumes**  $\exists (B::int) > 0. \exists c::nat \Rightarrow int.$   
 $(\forall_F n \text{ in sequentially. } B * b\ n = c\ n * a\ n - c(n+1) \wedge |c(n+1)| < a$   
 $n/2)$   
**shows**  $(\sum n. (b\ n / (\prod i \leq n. a\ i))) \in \mathbb{Q}$   
**proof** –  
**obtain**  $B::int$  **and**  $c::nat \Rightarrow int$  **and**  $N::nat$  **where**  $B > 0$  **and**  
 $large\text{-}n:\forall n \geq N. B * b\ n = c\ n * a\ n - c(n+1) \wedge real\text{-of-int } |c(n+1)| < a$   
 $n / 2$   
 $\wedge a\ n \geq 2$   
**proof** –  
**obtain**  $B\ c$  **where**  $B > 0$  **and**  $event1:\forall_F n \text{ in sequentially. } B * b\ n = c\ n * a$   
 $n - c(n+1)$   
 $\wedge real\text{-of-int } |c(n+1)| < real\text{-of-int } (a\ n) / 2$   
**using**  $assms$  **by**  $auto$   
**from**  $eventually\text{-conj}[OF\ event1\ a\text{-large,unfolding}\ eventually\text{-at-top-linorder}]$   
**obtain**  $N$  **where**  $\forall n \geq N. (B * b\ n = c\ n * a\ n - c(n+1)$   
 $\wedge real\text{-of-int } |c(n+1)| < real\text{-of-int } (a\ n) / 2) \wedge 2 \leq a\ n$   
**by**  $fastforce$   
**then show**  $?thesis$  **using**  $that[of\ B\ N\ c] \langle B > 0 \rangle$  **by**  $auto$   
**qed**  
**define**  $f$  **where**  $f = (\lambda n. real\text{-of-int } (b\ n) / real\text{-of-int } (prod\ a\ \{..n\}))$   
**define**  $S$  **where**  $S = (\sum n. f\ n)$   
**have**  $summable\ f$   
**unfolding**  $f\text{-def}$  **by**  $(rule\ aux\text{-series}\text{-summable})$   
**define**  $C$  **where**  $C = B * prod\ a\ \{..<N\} * (\sum n < N. f\ n)$   
**have**  $B * prod\ a\ \{..<N\} * S = C + real\text{-of-int } (c\ N)$   
**proof** –  
**define**  $h1$  **where**  $h1 \equiv (\lambda n. (c(n+N) * a(n+N)) / prod\ a\ \{N..n+N\})$   
**define**  $h2$  **where**  $h2 \equiv (\lambda n. c(n+N+1) / prod\ a\ \{N..n+N\})$   
**have**  $f\text{-}h12: B * prod\ a\ \{..<N\} * f(n+N) = h1\ n - h2\ n$  **for**  $n$   
**proof** –  
**define**  $g1$  **where**  $g1 \equiv (\lambda n. B * b(n+N))$   
**define**  $g2$  **where**  $g2 \equiv (\lambda n. prod\ a\ \{..<N\} / prod\ a\ \{..n + N\})$   
**have**  $B * prod\ a\ \{..<N\} * f(n+N) = (g1\ n * g2\ n)$   
**unfolding**  $f\text{-def}\ g1\text{-def}\ g2\text{-def}$  **by**  $(auto\ simp:algebra\text{-simps})$   
**moreover** **have**  $g1\ n = c(n+N) * a(n+N) - c(n+N+1)$   
**using**  $large\text{-}n[rule\text{-format,of}\ n+N]$  **unfolding**  $g1\text{-def}$  **by**  $auto$   
**moreover** **have**  $g2\ n = (1 / prod\ a\ \{N..n+N\})$   
**proof** –  
**have**  $prod\ a\ (\{..<N\} \cup \{N..n + N\}) = prod\ a\ \{..<N\} * prod\ a\ \{N..n +$   
 $N\}$   
**apply**  $(rule\ prod.\text{union-disjoint}[of\ \{..<N\}\ \{N..n+N\}\ a])$   
**by**  $auto$   
**moreover** **have**  $prod\ a\ (\{..<N\} \cup \{N..n + N\}) = prod\ a\ \{..n+N\}$   
**by**  $(simp\ add: inv\text{-disj-un-one}(4))$   
**ultimately show**  $?thesis$   
**unfolding**  $g2\text{-def}$   
**apply**  $simp$   
**using**  $a\text{-pos}$  **by**  $(metis\ less\text{-irrefl})$

**qed**  
**ultimately have**  $B*\text{prod } a \{..\lt N\} * f (n+N) = (c (n+N) * a (n+N) - c (n+N+1)) / \text{prod } a \{N..n+N\}$   
**by auto**  
**also have**  $\dots = h1\ n - h2\ n$   
**unfolding**  $h1\text{-def } h2\text{-def}$  **by**  $(\text{auto simp: algebra-simps diff-divide-distrib})$   
**finally show**  $?thesis$  **by simp**  
**qed**  
**have**  $B*\text{prod } a \{..\lt N\} * S = B*\text{prod } a \{..\lt N\} * ((\sum n \lt N. f\ n) + (\sum n. f (n+N)))$   
**using**  $\text{suminf-split-initial-segment}[OF \langle \text{summable } f \rangle, of\ N]$   
**unfolding**  $S\text{-def}$  **by**  $(\text{auto simp: algebra-simps})$   
**also have**  $\dots = C + B*\text{prod } a \{..\lt N\} * (\sum n. f (n+N))$   
**unfolding**  $C\text{-def}$  **by**  $(\text{auto simp: algebra-simps})$   
**also have**  $\dots = C + (\sum n. h1\ n - h2\ n)$   
**apply**  $(\text{subst suminf-mult}[symmetric])$   
**using**  $\langle \text{summable } f \rangle\ f\text{-h12}$  **by auto**  
**also have**  $\dots = C + h1\ 0$   
**proof**  $-$   
**have**  $(\lambda n. \sum i \leq n. h1\ i - h2\ i) \longrightarrow (\sum i. h1\ i - h2\ i)$   
**proof**  $(\text{rule summable-LIMSEQ'})$   
**have**  $(\lambda i. h1\ i - h2\ i) = (\lambda i. \text{real-of-int } (B * \text{prod } a \{..\lt N\}) * f (i + N))$   
**using**  $f\text{-h12}$  **by auto**  
**then show**  $\text{summable } (\lambda i. h1\ i - h2\ i)$   
**using**  $\langle \text{summable } f \rangle$  **by**  $(\text{simp add: summable-mult})$   
**qed**  
**moreover have**  $(\sum i \leq n. h1\ i - h2\ i) = h1\ 0 - h2\ n$  **for**  $n$   
**proof**  $(\text{induct } n)$   
**case**  $0$   
**then show**  $?case$  **by simp**  
**next**  
**case**  $(\text{Suc } n)$   
**have**  $(\sum i \leq \text{Suc } n. h1\ i - h2\ i) = (\sum i \leq n. h1\ i - h2\ i) + h1\ (n+1) - h2\ (n+1)$   
**by auto**  
**also have**  $\dots = h1\ 0 - h2\ n + h1\ (n+1) - h2\ (n+1)$  **using**  $\text{Suc}$  **by auto**  
**also have**  $\dots = h1\ 0 - h2\ (n+1)$   
**proof**  $-$   
**have**  $h2\ n = h1\ (n+1)$   
**unfolding**  $h2\text{-def } h1\text{-def}$   
**apply**  $(\text{auto simp: prod.nat-ivl-Suc'})$   
**using**  $a\text{-pos}$  **by**  $(\text{metis less-numeral-extra}(3))$   
**then show**  $?thesis$  **by auto**  
**qed**  
**finally show**  $?case$  **by simp**  
**qed**  
**ultimately have**  $(\lambda n. h1\ 0 - h2\ n) \longrightarrow (\sum i. h1\ i - h2\ i)$  **by simp**  
**then have**  $h2 \longrightarrow (h1\ 0 - (\sum i. h1\ i - h2\ i))$   
**apply**  $(\text{elim metric-tendsto-imp-tendsto})$

```

    by (auto intro!:eventuallyI simp add:dist-real-def)
  moreover have  $h2 \longrightarrow 0$ 
  proof -
    have  $h2-n:|h2\ n| < (1 / 2)^{\wedge}(n+1)$  for  $n$ 
    proof -
      have  $|h2\ n| = |c\ (n + N + 1)| / \text{prod } a\ \{N..n + N\}$ 
      unfolding  $h2\text{-def } \text{abs-divide}$ 
      using  $a\text{-pos}$  by (simp add:  $\text{abs-of-pos } \text{prod-pos}$ )
      also have  $\dots < (a\ (N+n) / 2) / \text{prod } a\ \{N..n + N\}$ 
      unfolding  $h2\text{-def}$ 
      apply (rule  $\text{divide-strict-right-mono}$ )
      subgoal using  $\text{large-}n[\text{rule-format,of } N+n]$  by (auto simp: $\text{algebra-simps}$ )
      subgoal using  $a\text{-pos}$  by (simp add:  $\text{prod-pos}$ )
      done
      also have  $\dots = 1 / (2 * \text{prod } a\ \{N..<n + N\})$ 
      apply (subst  $\text{ivl-disj-un}(6)[\text{of } N\ n+N,\text{symmetric}]$ )
      using  $a\text{-pos}[\text{rule-format,of } N+n]$  by (auto simp: $\text{algebra-simps}$ )
      also have  $\dots \leq (1/2)^{\wedge}(n+1)$ 
      proof (induct  $n$ )
        case 0
        then show ?case by auto
      next
        case (Suc  $n$ )
        define  $P$  where  $P=1 / \text{real-of-int } (2 * \text{prod } a\ \{N..<n + N\})$ 
        have  $1 / \text{real-of-int } (2 * \text{prod } a\ \{N..<\text{Suc } n + N\}) = P / a\ (n+N)$ 
        unfolding  $P\text{-def}$  by (auto simp:  $\text{prod.atLeastLessThan-Suc}$ )
        also have  $\dots \leq ((1 / 2)^{\wedge}(n + 1)) / a\ (n+N)$ 
        apply (rule  $\text{divide-right-mono}$ )
        subgoal unfolding  $P\text{-def}$  using  $\text{Suc}$  by auto
        subgoal by (simp add:  $a\text{-pos less-imp-le}$ )
        done
        also have  $\dots \leq ((1 / 2)^{\wedge}(n + 1)) / 2$ 
        apply (rule  $\text{divide-left-mono}$ )
        using  $\text{large-}n[\text{rule-format,of } n+N,\text{simplified}]$  by auto
        also have  $\dots = (1 / 2)^{\wedge}(n + 2)$  by auto
        finally show ?case by simp
      qed
      finally show ?thesis .
    qed
  have  $(\lambda n. (1 / 2)^{\wedge}(n+1)) \longrightarrow (0::\text{real})$ 
  using  $\text{tendsto-mult-right-zero}[OF\ \text{LIMSEQ-abs-realpow-zero2}[\text{of } 1/2,\text{simplified}],\ \text{of } 1/2]$ 
  by auto
  then show ?thesis
  apply (elim  $\text{Lim-null-comparison}[\text{rotated}]$ )
  using  $h2-n\ \text{less-eq-real-def}$  by (auto intro!:eventuallyI)
  qed
  ultimately have  $(\sum i. h1\ i - h2\ i) = h1\ 0$ 
  using  $\text{LIMSEQ-unique}$  by fastforce

```

**then show** *?thesis* **by** *simp*  
**qed**  
**also have**  $\dots = C + c N$   
**unfolding** *h1-def* **using** *a-pos*  
**by** *auto* (*metis less-irrefl*)  
**finally show** *?thesis* .  
**qed**  
**then have**  $S = (C + \text{real-of-int } (c N)) / (B * \text{prod } a \{..<N\})$   
**by** (*metis*  $\langle 0 < B \rangle$  *a-pos less-irrefl mult.commute mult-pos-pos*  
*nonzero-mult-div-cancel-right of-int-eq-0-iff prod-pos*)  
**moreover have**  $\dots \in \mathbb{Q}$   
**unfolding** *C-def f-def* **by** (*intro Rats-divide Rats-add Rats-mult Rats-of-int*  
*Rats-sum*)  
**ultimately show**  $S \in \mathbb{Q}$  **by** *auto*  
**qed**

**theorem** *theorem-2-1-Erdos-Straus* :  
 $(\sum n. (b n / (\prod i \leq n. a i))) \in \mathbb{Q} \longleftrightarrow (\exists (B::int)>0. \exists c::nat \Rightarrow \text{int.}$   
 $(\forall_F n \text{ in sequentially. } B * b n = c n * a n - c(n+1) \wedge |c(n+1)| < a n / 2))$   
**using** *ab-rationality-imp imp-ab-rational* **by** *auto*

The following is a Corollary to Theorem 2.1.

**corollary** *corollary-2-10-Erdos-Straus*:  
**assumes** *ab-event*: $\forall_F n \text{ in sequentially. } b n > 0 \wedge a (n+1) \geq a n$   
**and** *ba-lim-leq*: $\lim (\lambda n. (b(n+1) - b n) / a n) \leq 0$   
**and** *ba-lim-exist*:*convergent*  $(\lambda n. (b(n+1) - b n) / a n)$   
**and** *liminf*  $(\lambda n. a n / b n) = 0$   
**shows**  $(\sum n. (b n / (\prod i \leq n. a i))) \notin \mathbb{Q}$   
**proof**  
**assume**  $(\sum n. (b n / (\prod i \leq n. a i))) \in \mathbb{Q}$   
**then obtain**  $B c$  **where**  $B > 0$  **and** *abc-event*: $\forall_F n \text{ in sequentially. } B * b n = c$   
 $n * a n - c (n + 1)$   
 $\wedge |c (n + 1)| < a n / 2$  **and** *ca-vanish*: $(\lambda n. c (Suc n) / a n) \longrightarrow 0$   
**using** *ab-rationality-imp* **by** *auto*

**have** *bac-close*: $(\lambda n. B * b n / a n - c n) \longrightarrow 0$   
**proof** -  
**have**  $\forall_F n \text{ in sequentially. } B * b n - c n * a n + c (n + 1) = 0$   
**using** *abc-event* **by** (*auto elim!*:*eventually-mono*)  
**then have**  $\forall_F n \text{ in sequentially. } (B * b n - c n * a n + c (n+1)) / a n = 0$   
**apply** *eventually-elim*  
**by** *auto*  
**then have**  $\forall_F n \text{ in sequentially. } B * b n / a n - c n + c (n + 1) / a n = 0$   
**apply** *eventually-elim*  
**using** *a-pos* **by** (*auto simp:divide-simps*) (*metis less-irrefl*)  
**then have**  $(\lambda n. B * b n / a n - c n + c (n + 1) / a n) \longrightarrow 0$   
**by** (*simp add: eventually-mono tendsto-iff*)  
**from** *tendsto-diff*[*OF this ca-vanish*]  
**show** *?thesis* **by** *auto*

```

qed

have c-pos:  $\forall_F n$  in sequentially.  $c\ n > 0$ 
proof -
  from bac-close have *:  $\forall_F n$  in sequentially.  $c\ n \geq 0$ 
  apply (elim tendsto-of-int-diff-0)
  using ab-event a-large apply (eventually-elim)
  using  $\langle B > 0 \rangle$  by auto
  show ?thesis
  proof (rule ccontr)
    assume  $\neg (\forall_F n$  in sequentially.  $c\ n > 0)$ 
    moreover have  $\forall_F n$  in sequentially.  $c\ (Suc\ n) \geq 0 \wedge c\ n \geq 0$ 
      using * eventually-sequentially-Suc[of  $\lambda n. c\ n \geq 0$ ]
      by (metis (mono-tags, lifting) eventually-at-top-linorder le-Suc-eq)
    ultimately have  $\exists_F n$  in sequentially.  $c\ n = 0 \wedge c\ (Suc\ n) \geq 0$ 
      using eventually-elim2 frequently-def by fastforce
    moreover have  $\forall_F n$  in sequentially.  $b\ n > 0 \wedge B * b\ n = c\ n * a\ n - c$ 
      (n + 1)
      using ab-event abc-event by eventually-elim auto
    ultimately have  $\exists_F n$  in sequentially.  $c\ n = 0 \wedge c\ (Suc\ n) \geq 0 \wedge b\ n > 0$ 
       $\wedge B * b\ n = c\ n * a\ n - c\ (n + 1)$ 
      using frequently-eventually-frequently by fastforce
    from frequently-ex[OF this]
    obtain n where  $c\ n = 0 \wedge c\ (Suc\ n) \geq 0 \wedge b\ n > 0$ 
       $B * b\ n = c\ n * a\ n - c\ (n + 1)$ 
      by auto
    then have  $B * b\ n \leq 0$  by auto
    then show False using  $\langle b\ n > 0 \rangle \langle B > 0 \rangle$  using mult-pos-pos not-le by blast
  qed
qed

have bc-epsilon:  $\forall_F n$  in sequentially.  $b\ (n+1) / b\ n > (c\ (n+1) - \epsilon) / c\ n$ 
when  $\epsilon > 0 \wedge \epsilon < 1$  for  $\epsilon :: real$ 
proof -
  have  $\forall_F x$  in sequentially.  $|c\ (Suc\ x) / a\ x| < \epsilon / 2$ 
    using ca-vanish[unfolded tendsto-iff, rule-format, of  $\epsilon/2$ ]  $\langle \epsilon > 0 \rangle$  by auto
  moreover then have  $\forall_F x$  in sequentially.  $|c\ (x+2) / a\ (x+1)| < \epsilon / 2$ 
    apply (subst (asm) eventually-sequentially-Suc[symmetric])
    by simp
  moreover have  $\forall_F n$  in sequentially.  $B * b\ (n+1) = c\ (n+1) * a\ (n+1) -$ 
    c (n + 2)
    using abc-event
    apply (subst (asm) eventually-sequentially-Suc[symmetric])
    by (auto elim: eventually-mono)
  moreover have  $\forall_F n$  in sequentially.  $c\ n > 0 \wedge c\ (n+1) > 0 \wedge c\ (n+2) > 0$ 
  proof -
    have  $\forall_F n$  in sequentially.  $0 < c\ (Suc\ n)$ 
      using c-pos by (subst eventually-sequentially-Suc) simp
    moreover then have  $\forall_F n$  in sequentially.  $0 < c\ (Suc\ (Suc\ n))$ 

```

```

    using c-pos by (subst eventually-sequentially-Suc) simp
  ultimately show ?thesis using c-pos by eventually-elim auto
qed
ultimately show ?thesis using ab-event abc-event
proof eventually-elim
  case (elim n)
  define  $\varepsilon_0 \ \varepsilon_1$  where  $\varepsilon_0 = c \ (n+1) / a \ n$  and  $\varepsilon_1 = c \ (n+2) / a \ (n+1)$ 
  have  $\varepsilon_0 > 0 \ \varepsilon_1 > 0 \ \varepsilon_0 < \varepsilon/2 \ \varepsilon_1 < \varepsilon/2$  using a-pos elim by (auto simp:
 $\varepsilon_0$ -def  $\varepsilon_1$ -def)
  have  $(\varepsilon - \varepsilon_1) * c \ n > 0$ 
    using  $\langle \varepsilon_1 < \varepsilon / 2 \rangle$  elim(4) that(1) by auto
  moreover have  $\varepsilon_0 * (c \ (n+1) - \varepsilon) > 0$ 
    using  $\langle 0 < \varepsilon_0 \rangle$  elim(4) that(2) by auto
  ultimately have  $(\varepsilon - \varepsilon_1) * c \ n + \varepsilon_0 * (c \ (n+1) - \varepsilon) > 0$  by auto
  moreover have  $gt0: c \ n - \varepsilon_0 > 0$  using  $\langle \varepsilon_0 < \varepsilon / 2 \rangle$  elim(4) that(2) by
linarith
  moreover have  $c \ n > 0$  by (simp add: elim(4))
  ultimately have  $(c \ (n+1) - \varepsilon) / c \ n < (c \ (n+1) - \varepsilon_1) / (c \ n - \varepsilon_0)$ 
    by (auto simp: field-simps)
  also have  $\dots \leq (c \ (n+1) - \varepsilon_1) / (c \ n - \varepsilon_0) * (a \ (n+1) / a \ n)$ 
  proof -
    have  $(c \ (n+1) - \varepsilon_1) / (c \ n - \varepsilon_0) > 0$ 
      using  $gt0 \ \langle \varepsilon_1 < \varepsilon / 2 \rangle$  elim(4) that(2) by force
    moreover have  $a \ (n+1) / a \ n \geq 1$ 
      using a-pos elim(5) by auto
    ultimately show ?thesis by (metis mult-cancel-left1 mult-le-cancel-left-pos)
  qed
  also have  $\dots = (B * b \ (n+1)) / (B * b \ n)$ 
  proof -
    have  $B * b \ n = c \ n * a \ n - c \ (n + 1)$ 
      using elim by auto
    also have  $\dots = a \ n * (c \ n - \varepsilon_0)$ 
      using a-pos[rule-format,of n] unfolding  $\varepsilon_0$ -def by (auto simp:field-simps)
    finally have  $B * b \ n = a \ n * (c \ n - \varepsilon_0)$  .
    moreover have  $B * b \ (n+1) = a \ (n+1) * (c \ (n+1) - \varepsilon_1)$ 
      unfolding  $\varepsilon_1$ -def
      using a-pos[rule-format,of n+1]
      apply (subst  $\langle B * b \ (n + 1) = c \ (n + 1) * a \ (n + 1) - c \ (n + 2) \rangle$ )
      by (auto simp:field-simps)
    ultimately show ?thesis by (simp add: mult.commute)
  qed
  also have  $\dots = b \ (n+1) / b \ n$ 
    using  $\langle B > 0 \rangle$  by auto
  finally show ?case .
qed
qed

```

have  $eq-2-11: \exists_F n$  in sequentially.  $b \ (n+1) > b \ n + (1 - \varepsilon)^2 * a \ n / B$   
 when  $\varepsilon > 0 \ \varepsilon < 1 \ \neg (\forall_F n$  in sequentially.  $c \ (n+1) \leq c \ n)$  for  $\varepsilon :: real$

**proof** –  
**have**  $\exists_F x$  in sequentially.  $c x < c (Suc x)$  **using** *that(3)*  
**by** (*simp add:not-eventually-frequently-elim1*)  
**moreover have**  $\forall_F x$  in sequentially.  $|c (Suc x) / a x| < \varepsilon$   
**using** *ca-vanish[unfolded tendsto-iff,rule-format, of  $\varepsilon$ ]  $\langle\varepsilon>0$*  **by** *auto*  
**moreover have**  $\forall_F n$  in sequentially.  $c n > 0 \wedge c (n+1) > 0$   
**proof** –  
**have**  $\forall_F n$  in sequentially.  $0 < c (Suc n)$   
**using** *c-pos* **by** (*subst eventually-sequentially-Suc*) *simp*  
**then show** *?thesis* **using** *c-pos* **by** *eventually-elim auto*  
**qed**  
**ultimately show** *?thesis* **using** *ab-event abc-event bc-epsilon[OF  $\langle\varepsilon>0$ ]  $\langle\varepsilon<1$* ]

**proof** (*elim frequently-rev-mp,eventually-elim*)  
**case** (*elim n*)  
**then have**  $c (n+1) / a n < \varepsilon$   
**using** *a-pos[rule-format,of n]* **by** *auto*  
**also have**  $\dots \leq \varepsilon * c n$  **using** *elim(7) that(1)* **by** *auto*  
**finally have**  $c (n+1) / a n < \varepsilon * c n$  .  
**then have**  $c (n+1) / c n < \varepsilon * a n$   
**using** *a-pos[rule-format,of n] elim* **by** (*auto simp:field-simps*)  
**then have**  $(1 - \varepsilon) * a n < a n - c (n+1) / c n$   
**by** (*auto simp:algebra-simps*)  
**then have**  $(1 - \varepsilon)^2 * a n / B < (1 - \varepsilon) * (a n - c (n+1) / c n) / B$   
**apply** (*subst (asm) mult-less-cancel-right-pos[symmetric, of  $(1-\varepsilon)/B$ ]*)  
**using**  $\langle\varepsilon<1$   $\langle B>0$  **by** (*auto simp: divide-simps power2-eq-square mult-less-cancel-right-pos*)  
**then have**  $b n + (1 - \varepsilon)^2 * a n / B < b n + (1 - \varepsilon) * (a n - c (n+1) /$   
 $c n) / B$   
**using**  $\langle B>0$  **by** *auto*  
**also have**  $\dots = b n + (1 - \varepsilon) * ((c n * a n - c (n+1)) / c n) / B$   
**using** *elim* **by** (*auto simp:field-simps*)  
**also have**  $\dots = b n + (1 - \varepsilon) * (b n / c n)$   
**proof** –  
**have**  $B * b n = c n * a n - c (n+1)$  **using** *elim* **by** *auto*  
**from** *this[symmetric]* **show** *?thesis*  
**using**  $\langle B>0$  **by** *simp*  
**qed**  
**also have**  $\dots = (1 + (1-\varepsilon)/c n) * b n$   
**by** (*auto simp:algebra-simps*)  
**also have**  $\dots = ((c n + 1 - \varepsilon)/c n) * b n$   
**using** *elim* **by** (*auto simp:divide-simps*)  
**also have**  $\dots \leq ((c (n+1) - \varepsilon)/c n) * b n$   
**proof** –  
**define** *cp* **where**  $cp = c n + 1$   
**have**  $c (n+1) \geq cp$  **unfolding** *cp-def* **using**  $\langle c n < c (Suc n) \rangle$  **by** *auto*  
**moreover have**  $c n > 0$   $b n > 0$  **using** *elim* **by** *auto*  
**ultimately show** *?thesis*  
**apply** (*fold cp-def*)  
**by** (*auto simp:divide-simps*)

```

    qed
    also have ... < b (n+1)
      using elim by (auto simp:divide-simps)
    finally show ?case .
  qed
qed

have  $\forall_F n$  in sequentially.  $c (n+1) \leq c n$ 
proof (rule ccontr)
  assume  $\neg (\forall_F n$  in sequentially.  $c (n + 1) \leq c n$ )
  from eq-2-11[OF - - this,of 1/2]
  have  $\exists_F n$  in sequentially.  $b (n+1) > b n + 1/4 * a n / B$ 
    by (auto simp:algebra-simps power2-eq-square)
  then have  $*\exists_F n$  in sequentially.  $(b (n+1) - b n) / a n > 1 / (B * 4)$ 
    apply (elim frequently-elim1)
    subgoal for n
      using a-pos[rule-format,of n] by (auto simp:field-simps)
    done
  define f where  $f = (\lambda n. (b (n+1) - b n) / a n)$ 
  have  $f \longrightarrow \lim f$ 
    using convergent-LIMSEQ-iff ba-lim-exist unfolding f-def by auto
  from this[unfolded tendsto-iff,rule-format, of 1 / (B*4)]
  have  $\forall_F x$  in sequentially.  $|f x - \lim f| < 1 / (B * 4)$ 
    using  $\langle B > 0 \rangle$  by (auto simp:dist-real-def)
  moreover have  $\exists_F n$  in sequentially.  $f n > 1 / (B * 4)$ 
    using * unfolding f-def by auto
  ultimately have  $\exists_F n$  in sequentially.  $f n > 1 / (B * 4) \wedge |f n - \lim f| < 1 / (B * 4)$ 
    by (auto elim:frequently-eventually-frequently[rotated])
  from frequently-ex[OF this]
  obtain n where  $f n > 1 / (B * 4) \wedge |f n - \lim f| < 1 / (B * 4)$ 
    by auto
  moreover have  $\lim f \leq 0$  using ba-lim-leq unfolding f-def by auto
  ultimately show False by linarith
qed
then obtain N where N-dec: $\forall n \geq N. c (n+1) \leq c n$  by (meson eventually-at-top-linorder)
define max-c where  $\max-c = (\text{MAX } n \in \{..N\}. c n)$ 
have  $\max-c:c n \leq \max-c$  for n
proof (cases  $n \leq N$ )
  case True
    then show ?thesis unfolding max-c-def by simp
next
  case False
    then have  $n \geq N$  by auto
    then have  $c n \leq c N$ 
    proof (induct rule:nat-induct-at-least)
      case base
        then show ?case by simp
    next

```

```

    case (Suc n)
    then have  $c (n+1) \leq c n$  using N-dec by auto
    then show  $?case$  using  $\langle c n \leq c N \rangle$  by auto
  qed
  moreover have  $c N \leq \text{max-c}$  unfolding max-c-def by auto
  ultimately show ?thesis by auto
qed
have  $\text{max-c} > 0$ 
proof -
  obtain N where  $\forall n \geq N. 0 < c n$ 
  using c-pos[unfolded eventually-at-top-linorder] by auto
  then have  $c N > 0$  by auto
  then show ?thesis using max-c[of N] by simp
qed
have ba-limsup-bound:  $1/(B*(B+1)) \leq \text{limsup } (\lambda n. b n/a n)$ 
   $\text{limsup } (\lambda n. b n/a n) \leq \text{max-c} / B + 1 / (B+1)$ 
proof -
  define f where  $f = (\lambda n. b n/a n)$ 
  from tendsto-mult-right-zero[OF bac-close, of 1/B]
  have  $(\lambda n. f n - c n / B) \longrightarrow 0$ 
  unfolding f-def using  $\langle B > 0 \rangle$  by (auto simp: algebra-simps)
  from this[unfolded tendsto-iff, rule-format, of 1/(B+1)]
  have  $\forall_F x$  in sequentially.  $|f x - c x / B| < 1 / (B+1)$ 
  using  $\langle B > 0 \rangle$  by auto
  then have  $*:\forall_F n$  in sequentially.  $1/(B*(B+1)) \leq \text{ereal } (f n) \wedge \text{ereal } (f n) \leq$ 
 $\text{max-c} / B + 1 / (B+1)$ 
  using c-pos
proof eventually-elim
  case (elim n)
  then have  $f n - c n / B < 1 / (B+1)$  by auto
  then have  $f n < c n / B + 1 / (B+1)$  by simp
  also have  $\dots \leq \text{max-c} / B + 1 / (B+1)$ 
  using max-c[of n] using  $\langle B > 0 \rangle$  by (auto simp: divide-simps)
  finally have  $*:f n < \text{max-c} / B + 1 / (B+1)$  .

  have  $1/(B*(B+1)) = 1/B - 1 / (B+1)$ 
  using  $\langle B > 0 \rangle$  by (auto simp: divide-simps)
  also have  $\dots \leq c n/B - 1 / (B+1)$ 
  using  $\langle 0 < c n \rangle \langle B > 0 \rangle$  by (auto, auto simp: divide-simps)
  also have  $\dots < f n$  using elim by auto
  finally have  $1/(B*(B+1)) < f n$  .
  with  $*$  show ?case by simp
qed
show  $\text{limsup } f \leq \text{max-c} / B + 1 / (B+1)$ 
  apply (rule Limsup-bounded)
  using  $*$  by (auto elim: eventually-mono)
have  $1/(B*(B+1)) \leq \text{liminf } f$ 
  apply (rule Liminf-bounded)
  using  $*$  by (auto elim: eventually-mono)

```

also have  $\liminf f \leq \limsup f$  by (simp add: Liminf-le-Limsup)  
 finally show  $1/(B*(B+1)) \leq \limsup f$  .  
 qed

have  $0 < \text{inverse} (\text{ereal} (\text{max-c} / B + 1 / (B+1)))$   
 using  $\langle \text{max-c} > 0 \rangle \langle B > 0 \rangle$   
 by (simp add: pos-add-strict)  
 also have  $\dots \leq \text{inverse} (\limsup (\lambda n. b n/a n))$   
 proof (rule ereal-inverse-antimono[OF - ba-limsup-bound(2)])  
 have  $0 < 1/(B*(B+1))$  using  $\langle B > 0 \rangle$  by auto  
 also have  $\dots \leq \limsup (\lambda n. b n/a n)$  using ba-limsup-bound(1) .  
 finally show  $0 \leq \limsup (\lambda n. b n/a n)$  using zero-ereal-def by auto  
 qed

also have  $\dots = \liminf (\lambda n. \text{inverse} (\text{ereal} (b n/a n)))$   
 apply (subst Liminf-inverse-ereal[symmetric])  
 using a-pos ab-event by (auto elim!:eventually-mono simp:divide-simps)  
 also have  $\dots = \liminf (\lambda n. (a n/b n))$   
 apply (rule Liminf-eq)  
 using a-pos ab-event  
 apply (auto elim!:eventually-mono)  
 by (metis less-int-code(1))  
 finally have  $\liminf (\lambda n. (a n/b n)) > 0$  .  
 then show False using  $\langle \liminf (\lambda n. a n / b n) = 0 \rangle$  by simp  
 qed

end

### 3 Some auxiliary results on the prime numbers.

lemma nth-prime-nonzero[simp]:nth-prime  $n \neq 0$   
 by (simp add: prime-gt-0-nat prime-nth-prime)

lemma nth-prime-gt-zero[simp]:nth-prime  $n > 0$   
 by (simp add: prime-gt-0-nat prime-nth-prime)

lemma ratio-of-consecutive-primes:

$(\lambda n. \text{nth-prime} (n+1)/\text{nth-prime} n) \longrightarrow 1$

proof –

define f where  $f = (\lambda x. \text{real} (\text{nth-prime} (\text{Suc } x)) / \text{real} (\text{nth-prime } x))$

define g where  $g = (\lambda x. (\text{real } x * \ln (\text{real } x)) / (\text{real} (\text{Suc } x) * \ln (\text{real} (\text{Suc } x))))$

have p-n:  $(\lambda x. \text{real} (\text{nth-prime } x) / (\text{real } x * \ln (\text{real } x))) \longrightarrow 1$   
 using nth-prime-asymptotics[unfolded asymp-equiv-def,simplified] .

moreover have p-sn:  $(\lambda n. \text{real} (\text{nth-prime} (\text{Suc } n)) / (\text{real} (\text{Suc } n) * \ln (\text{real} (\text{Suc } n)))) \longrightarrow 1$   
 using nth-prime-asymptotics[unfolded asymp-equiv-def,simplified], THEN LIMSEQ-Suc] .

ultimately have  $(\lambda x. f x * g x) \longrightarrow 1$   
 using tendsto-divide[OF p-sn p-n]

```

  unfolding f-def g-def by (auto simp: algebra-simps)
moreover have g  $\longrightarrow$  1 unfolding g-def
  by real-asymp
ultimately have ( $\lambda x.$  if  $g\ x = 0$  then 0 else  $f\ x$ )  $\longrightarrow$  1
  apply (drule-tac tendsto-divide[OF - ⟨g  $\longrightarrow$  1⟩])
  by auto
then have f  $\longrightarrow$  1
proof (elim filterlim-mono-eventually)
  have  $\forall_F x$  in sequentially. (if  $g\ (x+3) = 0$  then 0
    else  $f\ (x+3) = f\ (x+3)$ )
    unfolding g-def by auto
  then show  $\forall_F x$  in sequentially. (if  $g\ x = 0$  then 0 else  $f\ x$ ) =  $f\ x$ 
    apply (subst (asm) eventually-sequentially-seg)
    by simp
qed auto
then show ?thesis unfolding f-def by auto
qed

lemma nth-prime-double-sqrt-less:
  assumes  $\varepsilon > 0$ 
  shows  $\forall_F n$  in sequentially. (nth-prime (2*n) - nth-prime n)
    / sqrt (nth-prime n) < n powr (1/2+ $\varepsilon$ )
proof -
  define pp ll where
    pp=( $\lambda n.$  (nth-prime (2*n) - nth-prime n) / sqrt (nth-prime n)) and
    ll=( $\lambda x::nat.$  x * ln x)
  have pp-pos: pp (n+1) > 0 for n
    unfolding pp-def by simp

  have ( $\lambda x.$  nth-prime (2 * x))  $\sim$ [sequentially] ( $\lambda x.$  (2 * x) * ln (2 * x))
    using nth-prime-asymptotics[THEN asymp-equiv-compose
      ,of (*) 2 sequentially,unfolded comp-def]
    using mult-nat-left-at-top pos2 by blast
  also have ...  $\sim$ [sequentially] ( $\lambda x.$  2 * x * ln x)
    by real-asymp
  finally have ( $\lambda x.$  nth-prime (2 * x))  $\sim$ [sequentially] ( $\lambda x.$  2 * x * ln x) .
  from this[unfolded asymp-equiv-def, THEN tendsto-mult-left,of 2]
  have ( $\lambda x.$  nth-prime (2 * x) / (x * ln x))  $\longrightarrow$  2
    unfolding asymp-equiv-def by auto
  moreover have *: ( $\lambda x.$  nth-prime x / (x * ln x))  $\longrightarrow$  1
    using nth-prime-asymptotics unfolding asymp-equiv-def by auto
  ultimately
  have ( $\lambda x.$  (nth-prime (2 * x) - nth-prime x) / ll x)  $\longrightarrow$  1
    unfolding ll-def
    apply -
    apply (drule (1) tendsto-diff)
    apply (subst of-nat-diff,simp)
    by (subst diff-divide-distrib,simp)
  moreover have ( $\lambda x.$  sqrt (nth-prime x) / sqrt (ll x))  $\longrightarrow$  1

```

**unfolding** *ll-def*  
**using** *tendsto-real-sqrt[OF \*]*  
**by** (*auto simp: real-sqrt-divide*)  
**ultimately have**  $(\lambda x. pp\ x * (sqrt\ (ll\ x) / (ll\ x))) \longrightarrow 1$   
**apply** –  
**apply** (*drule (1) tendsto-divide,simp*)  
**by** (*auto simp:field-simps of-nat-diff pp-def*)  
**moreover have**  $\forall_F x\ in\ sequentially. sqrt\ (ll\ x) / ll\ x = 1/sqrt\ (ll\ x)$   
**apply** (*subst eventually-sequentially-Suc[symmetric]*)  
**by** (*auto intro!:eventuallyI simp:ll-def divide-simps*)  
**ultimately have**  $(\lambda x. pp\ x / sqrt\ (ll\ x)) \longrightarrow 1$   
**apply** (*elim filterlim-mono-eventually*)  
**by** (*auto elim!:eventually-mono*) (*metis mult.right-neutral times-divide-eq-right*)  
**moreover have**  $(\lambda x. sqrt\ (ll\ x) / x\ powr\ (1/2+\epsilon)) \longrightarrow 0$   
**unfolding** *ll-def* **using**  $\langle \epsilon > 0 \rangle$  **by** *real-asymp*  
**ultimately have**  $(\lambda x. pp\ x / x\ powr\ (1/2+\epsilon) * (sqrt\ (ll\ x) / sqrt\ (ll\ x))) \longrightarrow 0$   
**apply** –  
**apply** (*drule (1) tendsto-mult*)  
**by** (*auto elim:filterlim-mono-eventually*)  
**moreover have**  $\forall_F x\ in\ sequentially. sqrt\ (ll\ x) / sqrt\ (ll\ x) = 1$   
**apply** (*subst eventually-sequentially-Suc[symmetric]*)  
**by** (*auto intro!:eventuallyI simp:ll-def*)  
**ultimately have**  $(\lambda x. pp\ x / x\ powr\ (1/2+\epsilon)) \longrightarrow 0$   
**apply** (*elim filterlim-mono-eventually*)  
**by** (*auto elim:eventually-mono*)  
**from** *tendstoD[OF this, of 1,simplified]*  
**show**  $\forall_F x\ in\ sequentially. pp\ x < x\ powr\ (1 / 2 + \epsilon)$   
**apply** (*elim eventually-mono-sequentially[of - 1]*)  
**using** *pp-pos* **by** *auto*  
**qed**

## 4 Theorem 3.1

Theorem 3.1 is an application of Theorem 2.1 with the sequences considered involving the prime numbers.

**theorem** *theorem-3-10-Erdos-Straus*:

**fixes** *a::nat*  $\Rightarrow$  *int*

**assumes** *a-pos*: $\forall n. a\ n > 0$  **and** *mono a*

**and** *nth-1*: $(\lambda n. nth\prime\ n / (a\ n)^{\wedge}2) \longrightarrow 0$

**and** *nth-2*:*liminf*  $(\lambda n. a\ n / nth\prime\ n) = 0$

**shows**  $(\sum n. (nth\prime\ n / (\prod i \leq n. a\ i))) \notin \mathbb{Q}$

**proof**

**assume** *asm*: $(\sum n. (nth\prime\ n / (\prod i \leq n. a\ i))) \in \mathbb{Q}$

**have** *a2-omega*: $(\lambda n. (a\ n)^{\wedge}2) \in \omega(\lambda x. x * ln\ x)$

**proof** –

**have**  $(\lambda n. real\ (nth\prime\ n)) \in o(\lambda n. real\ of\ int\ ((a\ n)^2))$

**apply** (*rule smalloI-tendsto*[*OF nth-1*])  
**using** *a-pos* **by** (*metis* (*mono-tags*, *lifting*) *less-int-code*(1)  
*not-eventuallyD of-int-0-eq-iff zero-eq-power2*)  
**moreover have** ( $\lambda x. \text{real } (nth\text{-prime } x) \in \Omega(\lambda x. \text{real } x * \ln (\text{real } x))$ )  
**using** *nth-prime-bigtheta*  
**by** *blast*  
**ultimately show** *?thesis*  
**using** *landau-omega.small-big-trans smallo-imp-smallomega* **by** *blast*  
**qed**

**have** *a-gt-1*: $\forall_F n$  *in sequentially*.  $1 < a \ n$   
**proof** –  
**have**  $\forall_F x$  *in sequentially*.  $|x * \ln x| \leq (a \ x)^2$   
**using** *a2-omega*[*unfolded smallomega-def,simplified,rule-format,of 1*]  
**by** *auto*  
**then have**  $\forall_F x$  *in sequentially*.  $|(x+3) * \ln (x+3)| \leq (a \ (x+3))^2$   
**apply** (*subst* (*asm*) *eventually-sequentially-seg*[*symmetric, of - 3*])  
**by** *simp*  
**then have**  $\forall_F n$  *in sequentially*.  $1 < a \ (n+3)$   
**proof** (*elim eventually-mono*)  
**fix** *x*  
**assume**  $|\text{real } (x + 3) * \ln (\text{real } (x + 3))| \leq \text{real-of-int } ((a \ (x + 3))^2)$   
**moreover have**  $|\text{real } (x + 3) * \ln (\text{real } (x + 3))| > 3$   
**proof** –  
**have**  $\ln (\text{real } (x + 3)) > 1$   
**using** *ln3-gt-1 ln-gt-1* **by** *force*  
**moreover have**  $\text{real}(x+3) \geq 3$  **by** *simp*  
**ultimately have**  $(x+3)*\ln (\text{real } (x + 3)) > 3*1$   
**by** (*smt* (*verit*, *best*) *mult-less-cancel-left1*)  
**then show** *?thesis* **by** *auto*  
**qed**  
**ultimately have**  $(a \ (x + 3))^2 > 3$   
**by** *linarith*  
**then show**  $1 < a \ (x + 3)$   
**by** (*smt* (*verit*) *assms*(1) *one-power2*)  
**qed**  
**then show** *?thesis*  
**using** *eventually-sequentially-seg*[*symmetric, of - 3*]  
**by** *blast*  
**qed**

**obtain** *B::int* **and** *c* **where**  
 $B > 0$  **and** *Bc-large*: $\forall_F n$  *in sequentially*.  $B * nth\text{-prime } n$   
 $= c \ n * a \ n - c \ (n + 1) \wedge |c \ (n + 1)| < a \ n / 2$   
**and** *ca-vanish*:  $(\lambda n. c \ (Suc \ n) / \text{real-of-int } (a \ n)) \longrightarrow 0$   
**proof** –  
**note** *a-gt-1*  
**moreover have**  $(\lambda n. \text{real-of-int } |int \ (nth\text{-prime } n)|$   
 $/ \text{real-of-int } (a \ (n - 1) * a \ n)) \longrightarrow 0$

```

proof –
  define f where  $f = (\lambda n. \text{nth-prime } (n+1) / (a\ n * a\ (n+1)))$ 
  define g where  $g = (\lambda n. 2 * \text{nth-prime } n / (a\ n)^2)$ 
  have  $\forall_F x$  in sequentially. norm (f x)  $\leq g\ x$ 
  proof –
    have  $\forall_F n$  in sequentially. nth-prime  $(n+1) < 2 * \text{nth-prime } n$ 
      using ratio-of-consecutive-primes[unfolded tendsto-iff
        ,rule-format, of 1, simplified]
      apply (elim eventually-mono)
      by (auto simp : divide-simps dist-norm)
    moreover have  $\forall_F n$  in sequentially. real-of-int  $(a\ n * a\ (n+1))$ 
       $\geq (a\ n)^2$ 
      apply (rule eventuallyI)
      using mono a by (auto simp: power2-eq-square a-pos incseq-SucD)
    ultimately show ?thesis unfolding f-def g-def
      apply eventually-elim
      apply (subst norm-divide)
      apply (rule-tac linordered-field-class.frac-le)
      using a-pos[rule-format, THEN order.strict-implies-not-eq]
      by auto
  qed
  moreover have g  $\longrightarrow 0$ 
    using nth-1[THEN tendsto-mult-right-zero, of 2] unfolding g-def
    by auto
  ultimately have f  $\longrightarrow 0$ 
    using Lim-null-comparison[of f g sequentially]
    by auto
  then show ?thesis
    unfolding f-def
    by (rule-tac LIMSEQ-imp-Suc) auto
  qed
  moreover have  $(\sum n. \text{real-of-int } (\text{int } (\text{nth-prime } n))$ 
     $/ \text{real-of-int } (\text{prod } a\ \{..n\})) \in \mathbb{Q}$ 
    using asm by simp
  ultimately have  $\exists B > 0. \exists c. (\forall_F n$  in sequentially.
     $B * \text{int } (\text{nth-prime } n) = c\ n * a\ n - c\ (n + 1) \wedge$ 
     $\text{real-of-int } |c\ (n + 1)| < \text{real-of-int } (a\ n) / 2) \wedge$ 
     $(\lambda n. \text{real-of-int } (c\ (\text{Suc } n)) / \text{real-of-int } (a\ n)) \longrightarrow 0$ 
    using ab-rationality-imp[OF a-pos, of nth-prime] by fast
  then show thesis
    apply clarify
    apply (rule-tac c=c and B=B in that)
    by auto
  qed

  have bac-close:  $(\lambda n. B * \text{nth-prime } n / a\ n - c\ n) \longrightarrow 0$ 
  proof –
    have  $\forall_F n$  in sequentially. B * nth-prime  $n - c\ n * a\ n + c\ (n + 1) = 0$ 
      using Bc-large by (auto elim!: eventually-mono)

```

**then have**  $\forall_F n$  *in sequentially.*  $(B * nth\text{-prime } n - c n * a n + c (n+1)) / a n = 0$   
**by** *eventually-elim auto*  
**then have**  $\forall_F n$  *in sequentially.*  $B * nth\text{-prime } n / a n - c n + c (n + 1) / a n = 0$   
**apply** *eventually-elim*  
**using** *a-pos* **by** *(auto simp:divide-simps) (metis less-irrefl)*  
**then have**  $(\lambda n. B * nth\text{-prime } n / a n - c n + c (n + 1) / a n) \longrightarrow 0$   
**by** *(simp add: eventually-mono tendsto-iff)*  
**from** *tendsto-diff[OF this ca-vanish]*  
**show** *?thesis* **by** *auto*  
**qed**

**have** *c-pos*: $\forall_F n$  *in sequentially.*  $c n > 0$

**proof** –

**from** *bac-close* **have**  $\forall_F n$  *in sequentially.*  $c n \geq 0$

**apply** *(elim tendsto-of-int-diff-0)*

**using** *a-gt-1* **apply** *(eventually-elim)*

**using**  $\langle B > 0 \rangle$  **by** *auto*

**show** *?thesis*

**proof** *(rule ccontr)*

**assume**  $\neg (\forall_F n$  *in sequentially.*  $c n > 0)$

**moreover have**  $\forall_F n$  *in sequentially.*  $c (Suc n) \geq 0 \wedge c n \geq 0$

**using**  $*$  *eventually-sequentially-Suc*[*of*  $\lambda n. c n \geq 0$ ]

**by** *(metis (mono-tags, lifting) eventually-at-top-linorder le-Suc-eq)*

**ultimately have**  $\exists_F n$  *in sequentially.*  $c n = 0 \wedge c (Suc n) \geq 0$

**using** *eventually-elim2 frequently-def* **by** *fastforce*

**moreover have**  $\forall_F n$  *in sequentially.*  $nth\text{-prime } n > 0$

$\wedge B * nth\text{-prime } n = c n * a n - c (n + 1)$

**using** *Bc-large* **by** *eventually-elim auto*

**ultimately have**  $\exists_F n$  *in sequentially.*  $c n = 0 \wedge c (Suc n) \geq 0$

$\wedge B * nth\text{-prime } n = c n * a n - c (n + 1)$

**using** *frequently-eventually-frequently* **by** *fastforce*

**from** *frequently-ex*[*OF this*]

**obtain**  $n$  **where**  $c n = 0 \wedge c (Suc n) \geq 0$

$B * nth\text{-prime } n = c n * a n - c (n + 1)$

**by** *auto*

**then have**  $B * nth\text{-prime } n \leq 0$  **by** *auto*

**then show** *False* **using**  $\langle B > 0 \rangle$

**by** *(simp add: mult-le-0-iff)*

**qed**

**qed**

**have** *B-nth-prime*: $\forall_F n$  *in sequentially.*  $nth\text{-prime } n > B$

**proof** –

**have**  $\forall_F x$  *in sequentially.*  $B+1 \leq nth\text{-prime } x$

**using** *nth-prime-at-top*[*unfolded filterlim-at-top-ge*[**where**  $c = nat B+1$ ],  
*rule-format, of nat B + 1, simplified*]

```

    apply (elim eventually-mono)
    using ⟨B>0⟩ by auto
  then show ?thesis
    by (auto elim: eventually-mono)
qed

have bc-epsilon:∀F n in sequentially. nth-prime (n+1)
  / nth-prime n > (c (n+1) - ε) / c n when ε>0 ε<1 for ε::real
proof -
  have ∀F x in sequentially. |c (Suc x) / a x| < ε / 2
    using ca-vanish[unfolding tendsto-iff, rule-format, of ε/2] ⟨ε>0 by auto
  moreover then have ∀F x in sequentially. |c (x+2) / a (x+1)| < ε / 2
    apply (subst (asm) eventually-sequentially-Suc[symmetric])
    by simp
  moreover have ∀F n in sequentially. B * nth-prime (n+1) = c (n+1) * a
    (n+1) - c (n + 2)
    using Bc-large
    apply (subst (asm) eventually-sequentially-Suc[symmetric])
    by (auto elim: eventually-mono)
  moreover have ∀F n in sequentially. c n > 0 ∧ c (n+1) > 0 ∧ c (n+2) > 0
proof -
  have ∀F n in sequentially. 0 < c (Suc n)
    using c-pos by (subst eventually-sequentially-Suc) simp
  moreover then have ∀F n in sequentially. 0 < c (Suc (Suc n))
    using c-pos by (subst eventually-sequentially-Suc) simp
  ultimately show ?thesis using c-pos by eventually-elim auto
qed
ultimately show ?thesis using Bc-large
proof eventually-elim
  case (elim n)
  define ε0 ε1 where ε0 = c (n+1) / a n and ε1 = c (n+2) / a (n+1)
  have ε0 > 0 ε1 > 0 ε0 < ε/2 ε1 < ε/2
    using a-pos elim ⟨mono a⟩
    by (auto simp: ε0-def ε1-def abs-of-pos)
  have (ε - ε1) * c n > 0
    using ⟨ε1 > 0⟩ ⟨ε1 < ε/2⟩ ⟨ε>0⟩ elim by auto
  moreover have A: ε0 * (c (n+1) - ε) > 0
    using ⟨ε0 > 0⟩ elim(4) that(2) by force
  ultimately have (ε - ε1) * c n + ε0 * (c (n+1) - ε) > 0 by auto
  moreover have B: c n - ε0 > 0 using ⟨ε0 < ε / 2⟩ elim(4) that(2) by
linarith
  moreover have c n > 0 by (simp add: elim(4))
  ultimately have (c (n+1) - ε) / c n < (c (n+1) - ε1) / (c n - ε0)
    by (auto simp: field-simps)
  also have ... ≤ (c (n+1) - ε1) / (c n - ε0) * (a (n+1) / a n)
proof -
  have (c (n+1) - ε1) / (c n - ε0) > 0
    using A ⟨0 < ε0⟩ B ⟨ε1 < ε / 2⟩ divide-pos-pos that(1) by force
  moreover have (a (n+1) / a n) ≥ 1

```

```

    using a-pos ⟨mono a⟩ by (simp add: mono-def)
    ultimately show ?thesis by (metis mult-cancel-left1 mult-le-cancel-left-pos)
  qed
  also have ... = (B * nth-prime (n+1)) / (B * nth-prime n)
  proof -
    have B * nth-prime n = c n * a n - c (n + 1)
      using elim by auto
    also have ... = a n * (c n - ε₀)
      using a-pos[rule-format,of n] unfolding ε₀-def by (auto simp:field-simps)
    finally have B * nth-prime n = a n * (c n - ε₀) .
    moreover have B * nth-prime (n+1) = a (n+1) * (c (n+1) - ε₁)
      unfolding ε₁-def
      using a-pos[rule-format,of n+1]
      apply (subst ⟨B * nth-prime (n + 1) = c (n + 1) * a (n + 1) - c (n +
2)⟩)
        by (auto simp:field-simps)
    ultimately show ?thesis by (simp add: mult.commute)
  qed
  also have ... = nth-prime (n+1) / nth-prime n
    using ⟨B>0⟩ by auto
  finally show ?case .
  qed
  qed

```

```

have c-ubound:∀ x. ∃ n. c n > x
proof (rule ccontr)
  assume ¬ (∀ x. ∃ n. x < c n)
  then obtain ub where ∀ n. c n ≤ ub ub > 0
    by (meson dual-order.trans int-one-le-iff-zero-less le-cases not-le)
  define pa where pa = (λ n. nth-prime n / a n)
  have pa-pos:∧ n. pa n > 0 unfolding pa-def by (simp add: a-pos)
  have liminf (λ n. 1 / pa n) = 0
    using nth-2 unfolding pa-def by auto
  then have (∃ y<ereal (real-of-int B / real-of-int (ub + 1))).
    ∃F x in sequentially. ereal (1 / pa x) ≤ y
    apply (subst less-Liminf-iff[symmetric])
    using ⟨0 < B⟩ ⟨0 < ub⟩ by auto
  then have ∃F x in sequentially. 1 / pa x < B/(ub+1)
    by (meson frequently-mono le-less-trans less-ereal.simps(1))
  then have ∃F x in sequentially. B*pa x > (ub+1)
    apply (elim frequently-elim1)
    by (metis ⟨0 < ub⟩ mult.left-neutral of-int-0-less-iff pa-pos pos-divide-less-eq
      pos-less-divide-eq times-divide-eq-left zless-add1-eq)
  moreover have ∀F x in sequentially. c x ≤ ub
    using ⟨∀ n. c n ≤ ub⟩ by simp
  ultimately have ∃F x in sequentially. B*pa x - c x > 1
    by (elim frequently-rev-mp eventually-mono) linarith
  moreover have (λ n. B * pa n - c n) → 0

```



**also have**  $\dots = (1 + (1 - \varepsilon) / c \ n) * \text{nth-prime } n$   
**by** *(auto simp: algebra-simps)*  
**also have**  $\dots = ((c \ n + 1 - \varepsilon) / c \ n) * \text{nth-prime } n$   
**using** *elim by (auto simp: divide-simps)*  
**also have**  $\dots \leq ((c \ (n + 1) - \varepsilon) / c \ n) * \text{nth-prime } n$   
**proof** –  
**define** *cp* **where**  $cp = c \ n + 1$   
**have**  $c \ (n + 1) \geq cp$  **unfolding** *cp-def* **using**  $\langle c \ n < c \ (n + 1) \rangle$  **by** *auto*  
**moreover have**  $c \ n > 0$  *nth-prime*  $n > 0$  **using** *elim by auto*  
**ultimately show** *?thesis*  
**apply** *(fold cp-def)*  
**by** *(auto simp: divide-simps)*  
**qed**  
**also have**  $\dots < \text{nth-prime } (n + 1)$   
**using** *elim by (auto simp: divide-simps)*  
**finally show**  $\text{real } (\text{nth-prime } n) + (1 - \varepsilon)^2 * \text{real-of-int } (a \ n)$   
 $/ \text{real-of-int } B < \text{real } (\text{nth-prime } (n + 1))$  .  
**qed**  
**qed**

**have** *c-neg-large*:  $\forall_F \ n$  *in sequentially*.  $c \ (n + 1) \neq c \ n$   
**proof** *(rule ccontr)*  
**assume**  $\neg (\forall_F \ n$  *in sequentially*.  $c \ (n + 1) \neq c \ n)$   
**then have** *that*:  $\exists_F \ n$  *in sequentially*.  $c \ (n + 1) = c \ n$   
**unfolding** *frequently-def* .  
**have**  $\forall_F \ x$  *in sequentially*.  $(B * \text{int } (\text{nth-prime } x) = c \ x * a \ x - c \ (x + 1))$   
 $\wedge |\text{real-of-int } (c \ (x + 1))| < \text{real-of-int } (a \ x) / 2) \wedge 0 < c \ x \wedge B < \text{int}$   
*(nth-prime x)*  
 $\wedge (c \ (x + 1) > c \ x \longrightarrow \text{nth-prime } (x + 1) > \text{nth-prime } x + a \ x / (2 * B))$   
**using** *Bc-large c-pos B-nth-prime eq-2-11 [of 1-1 / sqrt 2, simplified]*  
**by** *eventually-elim (auto simp: divide-simps)*  
**then have**  $\exists_F \ m$  *in sequentially*.  $\text{nth-prime } (m + 1) > (1 + 1 / (2 * B)) * \text{nth-prime}$   
*m*  
**proof** *(elim frequently-eventually-at-top [OF that, THEN frequently-at-top-elim])*  
**fix** *n*  
**assume**  $c \ (n + 1) = c \ n \wedge$   
 $(\forall y \geq n. (B * \text{int } (\text{nth-prime } y) = c \ y * a \ y - c \ (y + 1) \wedge$   
 $|\text{real-of-int } (c \ (y + 1))| < \text{real-of-int } (a \ y) / 2) \wedge$   
 $0 < c \ y \wedge B < \text{int } (\text{nth-prime } y) \wedge (c \ y < c \ (y + 1) \longrightarrow$   
 $\text{real } (\text{nth-prime } y) + \text{real-of-int } (a \ y) / \text{real-of-int } (2 * B)$   
 $< \text{real } (\text{nth-prime } (y + 1))))$   
**then have**  $c \ (n + 1) = c \ n$   
**and** *Bc-eq*:  $\forall y \geq n. B * \text{int } (\text{nth-prime } y) = c \ y * a \ y - c \ (y + 1) \wedge 0 < c \ y$   
 $\wedge |\text{real-of-int } (c \ (y + 1))| < \text{real-of-int } (a \ y) / 2$   
 $\wedge B < \text{int } (\text{nth-prime } y)$   
 $\wedge (c \ y < c \ (y + 1) \longrightarrow$   
 $\text{real } (\text{nth-prime } y) + \text{real-of-int } (a \ y) / \text{real-of-int } (2 * B)$   
 $< \text{real } (\text{nth-prime } (y + 1)))$   
**by** *auto*

```

obtain  $m$  where  $n < m$   $c\ m \leq c\ n$   $c\ n < c\ (m+1)$ 
proof –
  have  $\exists N. N > n \wedge c\ N > c\ n$ 
    using  $c$ -ubound[rule-format, of MAX  $x \in \{..n\}. c\ x$ ]
    by (metis Max-ge atMost-iff dual-order.trans finite-atMost finite-imageI
image-eqI
linorder-not-le order-refl)
  then obtain  $N$  where  $N > n$   $c\ N > c\ n$  by auto
  define  $A\ m$  where  $A = \{m. n < m \wedge (m+1) \leq N \wedge c\ (m+1) > c\ n\}$  and  $m$ 
= Min  $A$ 
  have finite  $A$  unfolding  $A$ -def
    by (metis (no-types, lifting)  $A$ -def add-leE finite-nat-set-iff-bounded-le
mem-Collect-eq)
  moreover have  $N-1 \in A$  unfolding  $A$ -def
    using  $\langle c\ n < c\ N \rangle \langle n < N \rangle \langle c\ (n+1) = c\ n \rangle$  nat-less-le by force
  ultimately have  $m \in A$ 
    using Min-in unfolding  $m$ -def by auto
  then have  $n < m$   $c\ n < c\ (m+1)$   $m > 0$ 
    unfolding  $m$ -def  $A$ -def by auto
  moreover have  $c\ m \leq c\ n$ 
proof (rule ccontr)
  assume  $\neg c\ m \leq c\ n$ 
  then have  $m-1 \in A$ 
    using  $\langle m \in A \rangle \langle c\ (n+1) = c\ n \rangle$  le-eq-less-or-eq less-diff-conv by (fastforce
simp:  $A$ -def)
  from Min-le[OF  $\langle$ finite  $A \rangle$  this, folded  $m$ -def]  $\langle m > 0 \rangle$  show False by auto
qed
ultimately show ?thesis using that[of  $m$ ] by auto
qed
have  $(1 + 1 / (2 * B)) * nth\prime\ m < nth\prime\ m + a\ m / (2 * B)$ 
proof –
  have  $nth\prime\ m < a\ m$ 
proof –
  have  $B * int\ (nth\prime\ m) < c\ m * (a\ m - 1)$ 
    using Bc-eq[rule-format, of  $m$ ]  $\langle c\ m \leq c\ n \rangle \langle c\ n < c\ (m+1) \rangle \langle n < m \rangle$ 
by (auto simp: algebra-simps)
  also have  $\dots \leq c\ n * (a\ m - 1)$ 
    by (simp add:  $\langle c\ m \leq c\ n \rangle$  a-pos mult-right-mono)
  finally have  $B * int\ (nth\prime\ m) < c\ n * (a\ m - 1)$  .
  moreover have  $c\ n \leq B$ 
proof –
  have  $B: B * int\ (nth\prime\ n) = c\ n * (a\ n - 1)$   $B < int\ (nth\prime\ n)$ 
    and  $c$ -a:  $|real\ of\ int\ (c\ (n+1))| < real\ of\ int\ (a\ n) / 2$ 
using Bc-eq[rule-format, of  $n$ ]  $\langle c\ (n+1) = c\ n \rangle$  by (auto simp: algebra-simps)
  from this(1) have  $c\ n\ dvd\ (B * int\ (nth\prime\ n))$ 
    by simp
  moreover have coprime  $(c\ n)$   $(int\ (nth\prime\ n))$ 
proof –
  have  $c\ n < int\ (nth\prime\ n)$ 

```

```

proof (rule ccontr)
  assume  $\neg c\ n < \text{int } (\text{nth-prime } n)$ 
  then have  $\text{asm}: c\ n \geq \text{int } (\text{nth-prime } n)$  by auto
  then have  $a\ n > 2 * \text{nth-prime } n$ 
    using  $c-a\ \langle c\ (n + 1) = c\ n \rangle$  by auto
  then have  $a\ n - 1 \geq 2 * \text{nth-prime } n$ 
    by simp
  then have  $a\ n - 1 > 2 * B$ 
    using  $\langle B < \text{int } (\text{nth-prime } n) \rangle$  by auto
  from mult-le-less-imp-less[OF asm this]  $\langle B > 0 \rangle$ 
  have  $\text{int } (\text{nth-prime } n) * (2 * B) < c\ n * (a\ n - 1)$ 
    by auto
  then show False using B
    by (smt (verit, best)  $\langle 0 < B \rangle$  mult.commute mult-right-mono)
qed
then have  $\neg \text{nth-prime } n\ \text{dvd } c\ n$ 
  by (simp add: Bc-eq zdvd-not-zless)
then have  $\text{coprime } (\text{int } (\text{nth-prime } n))\ (c\ n)$ 
  by (auto intro!: prime-imp-coprime-int)
then show ?thesis using coprime-commute by blast
qed
ultimately have  $c\ n\ \text{dvd } B$ 
  using coprime-dvd-mult-left-iff by auto
then show ?thesis using  $\langle 0 < B \rangle$  zdvd-imp-le by blast
qed
moreover have  $c\ n > 0$  using Bc-eq by blast
ultimately show ?thesis
  using  $\langle B > 0 \rangle$  by (smt (verit) a-pos mult-mono)
qed
then show ?thesis using  $\langle B > 0 \rangle$  by (auto simp: field-simps)
qed
also have  $\dots < \text{nth-prime } (m+1)$ 
  using Bc-eq[rule-format, of m]  $\langle n < m \rangle\ \langle c\ m \leq c\ n \rangle\ \langle c\ n < c\ (m+1) \rangle$ 
  by linarith
finally show  $\exists j > n. (1 + 1 / \text{real-of-int } (2 * B)) * \text{real } (\text{nth-prime } j)$ 
   $< \text{real } (\text{nth-prime } (j + 1))$  using  $\langle m > n \rangle$  by auto
qed
then have  $\exists_F m$  in sequentially. nth-prime (m+1)/nth-prime m > (1+1/(2*B))
  by (auto elim: frequently-elim1 simp: field-simps)
moreover have  $\forall_F m$  in sequentially. nth-prime (m+1)/nth-prime m <
 $(1+1/(2*B))$ 
  using ratio-of-consecutive-primes[unfolded tendsto-iff, rule-format, of 1/(2*B)]
   $\langle B > 0 \rangle$ 
  unfolding dist-real-def
  by (auto elim!: eventually-mono simp: algebra-simps)
ultimately show False by (simp add: eventually-mono frequently-def)
qed
have c-gt-half:  $\forall_F N$  in sequentially. card  $\{n \in \{N..<2*N\}. c\ n > c\ (n+1)\} >$ 

```

$N / 2$

**proof** –

**define**  $h$  **where**  $h = (\lambda n. (nth\text{-prime } (2 * n) - nth\text{-prime } n) / \text{sqrt } (nth\text{-prime } n))$

**have**  $\forall_F n$  *in sequentially*.  $h\ n < n / 2$

**proof** –

**have**  $\forall_F n$  *in sequentially*.  $h\ n < n\ \text{powr } (5/6)$

**using**  $nth\text{-prime-double-sqrt-less}$ [of 1/3]

**unfolding**  $h\text{-def}$  **by**  $auto$

**moreover** **have**  $\forall_F n$  *in sequentially*.  $n\ \text{powr } (5/6) < (n / 2)$

**by**  $real\text{-asympt}$

**ultimately** **show**  $?thesis$

**by**  $eventually\text{-elim } auto$

**qed**

**moreover** **have**  $\forall_F n$  *in sequentially*.  $\text{sqrt } (nth\text{-prime } n) / a\ n < 1 / (2 * B)$

**using**  $nth\text{-1}$ [ $THEN\ tendsto\text{-real-sqrt, unfolded } tendsto\text{-iff}$

$, rule\text{-format, of } 1 / (2 * B)$ ]  $\langle B > 0 \rangle\ a\text{-pos}$

**by**  $(auto\ simp: real\text{-sqrt-divide } abs\text{-of-pos})$

**ultimately** **have**  $\forall_F x$  *in sequentially*.  $c\ (x + 1) \neq c\ x$

$\wedge\ \text{sqrt } (nth\text{-prime } x) / a\ x < 1 / (2 * B)$

$\wedge\ h\ x < x / 2$

$\wedge\ (c\ (x + 1) > c\ x \longrightarrow nth\text{-prime } (x + 1) > nth\text{-prime } x + a\ x / (2 * B))$

**using**  $c\text{-neq-large } B\text{-nth-prime } eq\text{-2-11}$ [of 1-1 / sqrt 2, simplified]

**by**  $eventually\text{-elim } (auto\ simp: divide\text{-simps})$

**then** **show**  $?thesis$

**proof**  $(elim\ eventually\text{-at-top-mono})$

**fix**  $N$  **assume**  $N \geq 1$  **and**  $N\text{-asm}$ :  $\forall y \geq N. c\ (y + 1) \neq c\ y \wedge$

$\text{sqrt } (real\ (nth\text{-prime } y)) / real\text{-of-int } (a\ y)$

$< 1 / real\text{-of-int } (2 * B) \wedge h\ y < y / 2 \wedge$

$(c\ y < c\ (y + 1) \longrightarrow$

$real\ (nth\text{-prime } y) + real\text{-of-int } (a\ y) / real\text{-of-int } (2 * B)$

$< real\ (nth\text{-prime } (y + 1)))$

**define**  $S$  **where**  $S = \{n \in \{N.. < 2 * N\}. c\ n < c\ (n + 1)\}$

**define**  $g$  **where**  $g = (\lambda n. (nth\text{-prime } (n + 1) - nth\text{-prime } n) / \text{sqrt } (nth\text{-prime } n))$

**define**  $f$  **where**  $f = (\lambda n. nth\text{-prime } (n + 1) - nth\text{-prime } n)$

**have**  $g\text{-gt-1}$ :  $g\ n > 1$  **when**  $n \geq N$   $c\ n < c\ (n + 1)$  **for**  $n$

**proof** –

**have**  $nth\text{-prime } n + \text{sqrt } (nth\text{-prime } n) < nth\text{-prime } (n + 1)$

**proof** –

**have**  $nth\text{-prime } n + \text{sqrt } (nth\text{-prime } n) < nth\text{-prime } n + a\ n / (2 * B)$

**using**  $N\text{-asm}$ [ $rule\text{-format, OF } \langle n \geq N \rangle$ ]  $a\text{-pos}$

**by**  $(auto\ simp: field\text{-simps})$

**also** **have**  $\dots < nth\text{-prime } (n + 1)$

**using**  $N\text{-asm}$ [ $rule\text{-format, OF } \langle n \geq N \rangle$ ]  $\langle c\ n < c\ (n + 1) \rangle$  **by**  $auto$

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

**qed**

**then** **show**  $?thesis$  **unfolding**  $g\text{-def}$

```

    using ⟨c n < c (n + 1)⟩ by auto
qed
have g-geq-0: g n ≥ 0 for n
  unfolding g-def by auto

have finite S ∀ x ∈ S. x ≥ N ∧ c x < c (x + 1)
  unfolding S-def by auto
then have card S ≤ sum g S
proof (induct S)
  case empty
  then show ?case by auto
next
  case (insert x F)
  moreover have g x > 1
  proof -
    have c x < c (x + 1) x ≥ N using insert(4) by auto
    then show ?thesis using g-gt-1 by auto
  qed
  ultimately show ?case by simp
qed
also have ... ≤ sum g {N..<2*N}
  apply (rule sum-mono2)
  unfolding S-def using g-geq-0 by auto
also have ... ≤ sum (λn. f n / sqrt (nth-prime N)) {N..<2*N}
  unfolding f-def g-def by (auto intro!: sum-mono divide-left-mono)
also have ... = sum f {N..<2*N} / sqrt (nth-prime N)
  unfolding sum-divide-distrib[symmetric] by auto
also have ... = (nth-prime (2*N) - nth-prime N) / sqrt (nth-prime N)
proof -
  have sum f {N..<2 * N} = nth-prime (2 * N) - nth-prime N
proof (induct N)
  case 0
  then show ?case by simp
next
  case (Suc N)
  have ?case if N = 0
  proof -
    have sum f {Suc N..<2 * Suc N} = sum f {1}
      using that by (simp add: numeral-2-eq-2)
    also have ... = nth-prime 2 - nth-prime 1
      unfolding f-def by (simp add: numeral-2-eq-2)
    also have ... = nth-prime (2 * Suc N) - nth-prime (Suc N)
      using that by auto
    finally show ?thesis .
  qed
  moreover have ?case if N ≠ 0
  proof -
    have sum f {Suc N..<2 * Suc N} = sum f {N..<2 * Suc N} - f N
      apply (subst (2) sum.atLeast-Suc-lessThan)

```

using *that by auto*  
 also have ... =  $\text{sum } f \{N..<2 * N\} + f(2*N) + f(2*N+1) - f N$   
 by *auto*  
 also have ... =  $\text{nth-prime}(2 * \text{Suc } N) - \text{nth-prime}(\text{Suc } N)$   
 using *Suc unfolding f-def by auto*  
 finally show *?thesis .*  
 qed  
 ultimately show *?case by blast*  
 qed  
 then show *?thesis by auto*  
 qed  
 also have ... =  $h N$   
 unfolding *h-def by auto*  
 also have ... <  $N/2$   
 using *N-asm by auto*  
 finally have  $\text{card } S < N/2$  .

define *T* where  $T = \{n \in \{N..<2 * N\}. c n > c(n + 1)\}$   
 have  $T \cup S = \{N..<2 * N\}$   $T \cap S = \{\}$  *finite T*  
 unfolding *T-def S-def using N-asm by fastforce+*

then have  $\text{card } T + \text{card } S = \text{card } \{N..<2 * N\}$   
 using *card-Un-disjoint <finite S> bymetis*  
 also have ... =  $N$   
 by *simp*  
 finally have  $\text{card } T + \text{card } S = N$  .  
 with  $\langle \text{card } S < N/2 \rangle$   
 show  $\text{card } T > N/2$  by *linarith*

qed  
 qed

Inequality (3.5) in the original paper required a slight modification:

have *a-gt-plus*:  $\forall_F n$  in *sequentially*.  $c n > c(n+1) \longrightarrow a(n+1) > a n + (a n - c(n+1) - 1) / c(n+1)$

*proof -*

note *a-gt-1*[*THEN eventually-all-ge-at-top*] *c-pos*[*THEN eventually-all-ge-at-top*]

moreover have  $\forall_F n$  in *sequentially*.

$$B * \text{int}(\text{nth-prime}(n+1)) = c(n+1) * a(n+1) - c(n+2)$$

using *Bc-large*

apply (*subst(asm) eventually-sequentially-Suc[symmetric]*)

by (*auto elim:eventually-mono*)

moreover have  $\forall_F n$  in *sequentially*.

$$B * \text{int}(\text{nth-prime } n) = c n * a n - c(n+1) \wedge |c(n+1)|$$

<  $a n / 2$

using *Bc-large by (auto elim:eventually-mono)*

ultimately show *?thesis*

apply (*eventually-elim*)

proof (*rule impI*)

fix *n*

**assume**  $\forall y \geq n. 1 < a y \forall y \geq n. 0 < c y$   
**and**  
*Suc-n-eq*:  $B * \text{int } (\text{nth-prime } (n + 1)) = c (n + 1) * a (n + 1) - c (n + 1)$   
**2) and**  
 $B * \text{int } (\text{nth-prime } n) = c n * a n - c (n + 1) \wedge$   
 $\text{real-of-int } |c (n + 1)| < \text{real-of-int } (a n) / 2$   
**and**  $c (n + 1) < c n$   
**then have** *n-eq*:  $B * \text{int } (\text{nth-prime } n) = c n * a n - c (n + 1)$  **and**  
*c-less-a*:  $\text{real-of-int } |c (n + 1)| < \text{real-of-int } (a n) / 2$   
**by auto**  
**from**  $\langle \forall y \geq n. 1 < a y \rangle \langle \forall y \geq n. 0 < c y \rangle$   
**have**  $*: a n > 1 \ a (n+1) > 1 \ c n > 0$   
 $c (n+1) > 0 \ c (n+2) > 0$   
**by auto**  
**then have**  $(1+1/c (n+1)) * (a n - 1) / a (n+1) = (c (n+1)+1) * ((a n - 1) / (c (n+1) * a (n+1)))$   
**by** (*auto simp:field-simps*)  
**also have**  $\dots \leq c n * ((a n - 1) / (c (n+1) * a (n+1)))$   
**by** (*smt (verit) \*(4) <c (n + 1) < c n> a-pos divide-nonneg-nonneg mult-mono mult-nonneg-nonneg of-int-0-le-iff of-int-le-iff*)  
**also have**  $\dots = (c n * (a n - 1)) / (c (n+1) * a (n+1))$  **by auto**  
**also have**  $\dots < (c n * (a n - 1)) / (c (n+1) * a (n+1) - c (n+2))$   
**apply** (*rule divide-strict-left-mono*)  
**subgoal using**  $\langle c (n+2) > 0 \rangle$  **by auto**  
**unfolding** *Suc-n-eq[symmetric]* **using**  $* \langle B > 0 \rangle$  **by auto**  
**also have**  $\dots < (c n * a n - c (n+1)) / (c (n+1) * a (n+1) - c (n+2))$   
**apply** (*rule frac-less*)  
**unfolding** *Suc-n-eq[symmetric]* **using**  $* \langle B > 0 \rangle \langle c (n + 1) < c n \rangle$   
**by** (*auto simp:algebra-simps*)  
**also have**  $\dots = \text{nth-prime } n / \text{nth-prime } (n+1)$   
**unfolding** *Suc-n-eq[symmetric]* *n-eq[symmetric]* **using**  $\langle B > 0 \rangle$  **by auto**  
**also have**  $\dots < 1$  **by auto**  
**finally have**  $(1 + 1 / \text{real-of-int } (c (n + 1))) * \text{real-of-int } (a n - 1) / \text{real-of-int } (a (n + 1)) < 1$  .  
**then show**  $a n + (a n - c (n + 1) - 1) / (c (n + 1)) < (a (n + 1))$   
**using**  $*$  **by** (*auto simp:field-simps*)  
**qed**  
**qed**  
**have** *a-gt-1*:  $\forall_F n$  *in sequentially*.  $c n > c (n+1) \longrightarrow a (n+1) > a n + 1$   
**using** *Bc-large a-gt-plus c-pos[THEN eventually-all-ge-at-top]*  
**apply** *eventually-elim*  
**proof** (*rule impI*)  
**fix**  $n$  **assume**  
 $c (n + 1) < c n \longrightarrow a n + (a n - c (n + 1) - 1) / c (n + 1) < a (n + 1)$   
**1)**  
 $c (n + 1) < c n$  **and** *B-eq*:  $B * \text{int } (\text{nth-prime } n) = c n * a n - c (n + 1) \wedge$   
 $|\text{real-of-int } (c (n + 1))| < \text{real-of-int } (a n) / 2$  **and** *c-pos*:  $\forall y \geq n. 0 < c y$   
**from** *this*(1,2)  
**have**  $a n + (a n - c (n + 1) - 1) / c (n + 1) < a (n + 1)$  **by auto**

```

moreover have  $a\ n - 2 * c\ (n+1) > 0$ 
  using B-eq c-pos[rule-format,of n+1] by auto
then have  $a\ n - 2 * c\ (n+1) \geq 1$  by simp
then have  $(a\ n - c\ (n + 1) - 1) / c\ (n + 1) \geq 1$ 
  using c-pos[rule-format,of n+1] by (auto simp:field-simps)
ultimately show  $a\ n + 1 < a\ (n + 1)$  by auto
qed

```

The following corresponds to inequality (3.6) in the paper, which had to be slightly corrected:

```

have a-gt-sqrt: $\forall_F\ n\ \text{in}\ \text{sequentially. } c\ n > c\ (n+1) \longrightarrow a\ (n+1) > a\ n + (\text{sqrt}\ n - 2)$ 
proof -
  have a-2N: $\forall_F\ N\ \text{in}\ \text{sequentially. } a\ (2*N) \geq N / 2 + 1$ 
    using c-gt-half a-gt-1[THEN eventually-all-ge-at-top]
  proof eventually-elim
    case (elim N)
    define S where  $S = \{n \in \{N..<2 * N\}. c\ (n + 1) < c\ n\}$ 
    define f where  $f = (\lambda n. a\ (Suc\ n) - a\ n)$ 

    have f-1: $\forall x \in S. f\ x \geq 1$  and f-0: $\forall x. f\ x \geq 0$ 
      subgoal using elim unfolding S-def f-def by auto
      subgoal using  $\langle \text{mono } a \rangle$ [THEN incseq-SucD] unfolding f-def by auto
      done
    have  $N / 2 < \text{card } S$ 
      using elim unfolding S-def by auto
    also have  $\dots \leq \text{sum } f\ S$ 
      unfolding of-int-sum
      apply (rule sum-bounded-below[of - 1,simplified])
      using f-1 by auto
    also have  $\dots \leq \text{sum } f\ \{N..<2 * N\}$ 
      unfolding of-int-sum
      apply (rule sum-mono2)
      unfolding S-def using f-0 by auto
    also have  $\dots = a\ (2*N) - a\ N$ 
      unfolding of-int-sum f-def of-int-diff
      apply (rule sum-Suc-diff')
      by auto
    finally have  $N / 2 < a\ (2*N) - a\ N .$ 
    then show ?case using a-pos[rule-format,of N] by linarith
qed

```

```

have a-n4: $\forall_F\ n\ \text{in}\ \text{sequentially. } a\ n > n/4$ 
proof -
  obtain N where  $a\ N : \forall n \geq N. a\ (2*n) \geq n / 2 + 1$ 
    using a-2N unfolding eventually-at-top-linorder by auto
  have  $a\ n > n/4$  when  $n \geq 2*N$  for n
  proof -
    define n' where  $n' = n\ \text{div } 2$ 

```

```

have  $n' \geq N$  unfolding  $n'$ -def using that by auto
have  $n/4 < n'/2 + 1$ 
  unfolding  $n'$ -def by auto
also have  $\dots \leq a (2 * n')$ 
  using  $a-N \langle n' \geq N \rangle$  by auto
also have  $\dots \leq a n$  unfolding  $n'$ -def
  apply (cases even n)
  subgoal by simp
  subgoal by (simp add: assms(2) incseqD)
  done
finally show ?thesis .
qed
then show ?thesis
  unfolding eventually-at-top-linorder by auto
qed

have c-sqrt:  $\forall_F n$  in sequentially.  $c n < \text{sqrt } n / 4$ 
proof -
  have  $\forall_F x$  in sequentially.  $x > 1$  by simp
  moreover have  $\forall_F x$  in sequentially.  $\text{real } (nth\text{-prime } x) / (\text{real } x * \ln (\text{real } x)) < 2$ 
  using nth-prime-asymptotics[unfolded asymp-equiv-def, THEN order-tendstoD(2), of 2]
  by simp
  ultimately have  $\forall_F n$  in sequentially.  $c n < B * 8 * \ln n + 1$  using a-n/4
Bc-large
proof eventually-elim
  case (elim n)
  from this(4) have  $c n = (B * nth\text{-prime } n + c (n + 1)) / a n$ 
    using a-pos[rule-format, of n]
    by (auto simp: divide-simps)
  also have  $\dots = (B * nth\text{-prime } n) / a n + c (n + 1) / a n$ 
    by (auto simp: divide-simps)
  also have  $\dots < (B * nth\text{-prime } n) / a n + 1$ 
  proof -
    have  $c (n + 1) / a n < 1$  using elim(4) by auto
    then show ?thesis by auto
  qed
also have  $\dots < B * 8 * \ln n + 1$ 
proof -
  have  $B * nth\text{-prime } n < 2 * B * n * \ln n$ 
    using  $\langle \text{real } (nth\text{-prime } n) / (\text{real } n * \ln (\text{real } n)) < 2 \rangle \langle B > 0 \rangle \langle 1 < n \rangle$ 
    by (auto simp: divide-simps)
  moreover have  $\text{real } n / 4 < \text{real-of-int } (a n)$  by fact
  ultimately have  $(B * nth\text{-prime } n) / a n < (2 * B * n * \ln n) / (n / 4)$ 
    apply (rule-tac frac-less)
    using  $\langle B > 0 \rangle \langle 1 < n \rangle$  by auto
  also have  $\dots = B * 8 * \ln n$ 
    using  $\langle 1 < n \rangle$  by auto

```

```

    finally show ?thesis by auto
  qed
  finally show ?case .
  qed
  moreover have  $\forall_F n$  in sequentially.  $B*8 *ln n + 1 < sqrt n / 4$ 
    by real-asymp
  ultimately show ?thesis
    by eventually-elim auto
  qed

have
   $\forall_F n$  in sequentially.  $0 < c (n+1)$ 
   $\forall_F n$  in sequentially.  $c (n+1) < sqrt (n+1) / 4$ 
   $\forall_F n$  in sequentially.  $n > 4$ 
   $\forall_F n$  in sequentially.  $(n - 4) / sqrt (n + 1) + 1 > sqrt n$ 
  subgoal using c-pos[THEN eventually-all-ge-at-top]
    by eventually-elim auto
  subgoal using c-sqrt[THEN eventually-all-ge-at-top]
    by eventually-elim (use le-add1 in blast)
  subgoal by simp
  subgoal
    by real-asymp
  done
  then show ?thesis using a-gt-plus a-n4
    apply eventually-elim
  proof (rule impI)
    fix n assume asm:  $0 < c (n + 1)$   $c (n + 1) < sqrt (real (n + 1)) / 4$  and
      a-ineq:  $c (n + 1) < c n \longrightarrow a n + (a n - c (n + 1) - 1) / c (n + 1) <$ 
      a (n + 1)
       $c (n + 1) < c n$  and  $n / 4 < a n$   $n > 4$ 
      and n-neq:  $sqrt (real n) < real (n - 4) / sqrt (real (n + 1)) + 1$ 

    have  $(n-4) / sqrt(n+1) = (n/4 - 1) / (sqrt (real (n + 1)) / 4)$ 
      using  $\langle n > 4 \rangle$  by (auto simp: divide-simps)
    also have  $\dots < (a n - 1) / c (n + 1)$ 
      apply (rule frac-less)
      using  $\langle n > 4 \rangle$   $\langle n / 4 < a n \rangle$   $\langle 0 < c (n + 1) \rangle$   $\langle c (n + 1) < sqrt (real (n$ 
      + 1)) / 4 \rangle
      by auto
    also have  $\dots - 1 = (a n - c (n + 1) - 1) / c (n + 1)$ 
      using  $\langle 0 < c (n + 1) \rangle$  by (auto simp: field-simps)
    also have  $a n + \dots < a (n+1)$ 
      using a-ineq by auto
    finally have  $a n + ((n - 4) / sqrt (n + 1) - 1) < a (n + 1)$  by simp
    moreover have  $(n - 4) / sqrt (n + 1) - 1 > sqrt n - 2$ 
      using n-neq[THEN diff-strict-right-mono, of 2]  $\langle n > 4 \rangle$ 
      by (auto simp: algebra-simps of-nat-diff)
    ultimately show  $real-of-int (a n) + (sqrt (real n) - 2) < real-of-int (a (n$ 
    + 1))
```

by argo  
qed  
qed

The following corresponds to inequality  $a_{2N} > N^{3/2}/2$  in the paper, which had to be slightly corrected:

have  $a_{2N}$ -sqr:  $\forall F N$  in sequentially.  $a (2*N) > \text{real } N * (\text{sqrt } (\text{real } N)/2 - 1)$   
using  $c$ -gt-half  $a$ -gt-sqrt[THEN eventually-all-ge-at-top] eventually-gt-at-top[of 4]

proof eventually-elim

case (elim  $N$ )

define  $S$  where  $S = \{n \in \{N..<2 * N\}. c (n + 1) < c n\}$

define  $f$  where  $f = (\lambda n. a (Suc n) - a n)$

have  $f$ - $N$ :  $\forall x \in S. f x \geq \text{sqrt } N - 2$

proof

fix  $x$  assume  $x \in S$

then have  $\text{sqrt } (\text{real } x) - 2 < f x$   $x \geq N$

using elim unfolding  $S$ -def  $f$ -def by auto

moreover have  $\text{sqrt } x - 2 \geq \text{sqrt } N - 2$

using  $\langle x \geq N \rangle$  by simp

ultimately show  $\text{sqrt } (\text{real } N) - 2 \leq \text{real-of-int } (f x)$  by argo

qed

have  $f$ -0:  $\forall x. f x \geq 0$

using  $\langle \text{mono } a \rangle$ [THEN incseq-SucD] unfolding  $f$ -def by auto

have  $(N / 2) * (\text{sqrt } N - 2) < \text{card } S * (\text{sqrt } N - 2)$

apply (rule mult-strict-right-mono)

subgoal using elim unfolding  $S$ -def by auto

subgoal using  $\langle N > 4 \rangle$

by (metis diff-gt-0-iff-gt numeral-less-real-of-nat-iff real-sqrt-four real-sqrt-less-iff)

done

also have  $\dots \leq \text{sum } f S$

unfolding of-int-sum

apply (rule sum-bounded-below)

using  $f$ - $N$  by auto

also have  $\dots \leq \text{sum } f \{N..<2 * N\}$

unfolding of-int-sum

apply (rule sum-mono2)

unfolding  $S$ -def using  $f$ -0 by auto

also have  $\dots = a (2*N) - a N$

unfolding of-int-sum  $f$ -def of-int-diff

apply (rule sum-Suc-diff')

by auto

finally have  $\text{real } N / 2 * (\text{sqrt } (\text{real } N) - 2) < \text{real-of-int } (a (2 * N) - a N)$

then have  $\text{real } N / 2 * (\text{sqrt } (\text{real } N) - 2) < a (2 * N)$

using  $a$ -pos[rule-format,of  $N$ ] by linarith

**then show** *?case* **by** (*auto simp:field-simps*)  
**qed**

The following part is required to derive the final contradiction of the proof.

**have** *a-n-sqrt*: $\forall_F n$  *in sequentially*.  $a n > (((n-1)/2) \text{ powr } (3/2) - (n-1)) / 2$   
**proof** (*rule sequentially-even-odd-imp*)  
**define** *f* **where**  $f = (\lambda N. ((\text{real } (2 * N - 1) / 2) \text{ powr } (3 / 2) - \text{real } (2 * N - 1)) / 2)$   
**define** *g* **where**  $g = (\lambda N. \text{real } N * (\text{sqrt } (\text{real } N) / 2 - 1))$   
**have**  $\forall_F N$  *in sequentially*.  $g N > f N$   
**unfolding** *f-def g-def*  
**by** *real-asymp*  
**moreover have**  $\forall_F N$  *in sequentially*.  $a (2 * N) > g N$   
**unfolding** *g-def* **using** *a-2N-sqrt* .  
**ultimately show**  $\forall_F N$  *in sequentially*.  $f N < a (2 * N)$   
**by** *eventually-elim auto*

**next**

**define** *f* **where**  $f = (\lambda N. ((\text{real } (2 * N + 1 - 1) / 2) \text{ powr } (3 / 2) - \text{real } (2 * N + 1 - 1)) / 2)$   
**define** *g* **where**  $g = (\lambda N. \text{real } N * (\text{sqrt } (\text{real } N) / 2 - 1))$   
**have**  $\forall_F N$  *in sequentially*.  $g N = f N$   
**using** *eventually-gt-at-top[of 0]*  
**apply** *eventually-elim*  
**unfolding** *f-def g-def*  
**by** (*auto simp:algebra-simps powr-half-sqrt[symmetric] powr-mult-base*)  
**moreover have**  $\forall_F N$  *in sequentially*.  $a (2 * N) > g N$   
**unfolding** *g-def* **using** *a-2N-sqrt* .  
**moreover have**  $\forall_F N$  *in sequentially*.  $a (2 * N + 1) \geq a (2 * N)$   
**apply** (*rule eventuallyI*)  
**using**  $\langle \text{mono } a \rangle$  **by** (*simp add: incseqD*)  
**ultimately show**  $\forall_F N$  *in sequentially*.  $f N < (a (2 * N + 1))$   
**by** *eventually-elim auto*

**qed**

**have** *a-nth-prime-gt*: $\forall_F n$  *in sequentially*.  $a n / \text{nth-prime } n > 1$

**proof** –

**define** *f* **where**  $f = (\lambda n::\text{nat}. (((n-1)/2) \text{ powr } (3/2) - (n-1)) / 2)$   
**have**  $\forall_F x$  *in sequentially*.  $\text{real } (\text{nth-prime } x) / (\text{real } x * \ln (\text{real } x)) < 2$   
**using** *nth-prime-asymptotics[unfolded asymp-equiv-def, THEN order-tendstoD(2), of 2]*  
**by** *simp*  
**from** *this eventually-gt-at-top[of 1]*  
**have**  $\forall_F n$  *in sequentially*.  $\text{real } (\text{nth-prime } n) < 2 * (\text{real } n * \ln n)$   
**by** *eventually-elim (auto simp:field-simps)*  
**moreover have**  $\forall_F N$  *in sequentially*.  $f N > 0$   
**unfolding** *f-def*  
**by** *real-asymp*  
**moreover have**  $\forall_F n$  *in sequentially*.  $f n < a n$

```

    using a-n-sqrt unfolding f-def .
  ultimately have  $\forall_F n$  in sequentially.  $a n / \text{nth-prime } n > f n / (2*(\text{real } n * \ln n))$ 
  proof eventually-elim
    case (elim n)
    then show ?case
      by (auto intro: frac-less2)
  qed
  moreover have  $\forall_F n$  in sequentially.  $(f n) / (2*(\text{real } n * \ln n)) > 1$ 
  unfolding f-def by real-asymp
  ultimately show ?thesis
    by eventually-elim argo
  qed

  have a-nth-prime-lt: $\exists_F n$  in sequentially.  $a n / \text{nth-prime } n < 1$ 
  proof -
    have liminf  $(\lambda x. a x / \text{nth-prime } x) < 1$ 
      using nth-2 by auto
    from this[unfolded less-Liminf-iff]
    show ?thesis
      by (smt (verit) ereal-less(3) frequently-elim1 le-less-trans)
  qed

  from a-nth-prime-gt a-nth-prime-lt show False
  by (simp add: eventually-mono frequently-def)
  qed

```

## 5 Acknowledgements

A.K.-A. and W.L. were supported by the ERC Advanced Grant ALEXANDRIA (Project 742178) funded by the European Research Council and led by Professor Lawrence Paulson at the University of Cambridge, UK.

end

## References

- [1] P. Erdős and E. Straus. On the irrationality of certain series. *Pacific journal of mathematics*, 55(1):85–92, 1974.