Irrational Rapidly Convergent Series

Angeliki Koutsoukou-Argyraki and Wenda Li

August 16, 2018

Abstract

We formalize with Isabelle/HOL a proof of a theorem by J. Hančl asserting the irrationality of the sum of a series consisting of rational numbers, built up by sequences that fulfill certain properties. Even though the criterion is a number theoretic result, the proof makes use only of analytical arguments. We also formalize a corollary of the theorem for a specific series fulfilling the assumptions of the theorem.

Contents

1 Main Theorem and Sketch of the Proof 1

2 Corollary 2

3 Irrational Rapidly Convergent Series 3
   3.1 Misc ................................................................. 3
   3.2 Auxiliary lemmas and the main proof ......................... 6

4 Acknowledgements 27

1 Main Theorem and Sketch of the Proof

We formalize the proof of the following theorem by J. Hančl (Theorem 3 in [1]) :

Theorem 1. (Theorem 3 in [1]) Let $A \in \mathbb{R}$ with $A > 1$. Let $\{d_n\}_{n=1}^{\infty} \in \mathbb{R}$ with $d_n > 1$ for all $n \in \mathbb{N}$. Let $\{a_n\}_{n=1}^{\infty} \in \mathbb{Z}^+$, $\{b_n\}_{n=1}^{\infty} \in \mathbb{Z}^+$ such that :

(1) $\lim_{n \to \infty} \frac{1}{a_n} = A,$

for all sufficiently large $n \in \mathbb{N}$ :

(2) $\frac{A}{a_n^{\frac{1}{d_n}}} > \prod_{j=n}^{\infty} d_j$
and

\[(3) \lim_{n \to \infty} d_n^{a_n} = \infty.\]

Then the series \(\alpha = \sum_{n=1}^{\infty} \frac{b_n}{a_n}\) is an irrational number.

The first step is to show that the series \(\sum_{n=1}^{\infty} \frac{b_n}{a_n}\) converges to some \(\alpha \in \mathbb{R}\). To show that \(\alpha \in \mathbb{R} \setminus \mathbb{Q}\) we argue by proof by contradiction (to this end several auxiliary lemmas are firstly shown). In particular, assuming that \(\alpha \in \mathbb{Q}\), i.e. that there exist \(p, q \in \mathbb{Z}^+\) such that \(\alpha = \frac{p}{q}\), we show that a quantity \(A(n) \geq 1\) for all \(n \in \mathbb{N}\). At the same time, we find \(n \in \mathbb{N}\) for which \(A(n) < 1\), yielding a contradiction from which we deduce the irrationality of the sum of the series.

For the proof see [1]. We note that the proof involves only elementary Analysis (criteria for convergence/divergence for sequences and series and several inequalities) and not any arithmetical/number theoretic arguments. Obviously for the formal proof we had to make many intermediate arguments explicit. Proofs of length of roughly 2 A4 pages in the original paper by J. Hančl were formalized in almost 1100 lines of code.

2 Corollary

We moreover formalize the following corollary that asserts the irrationality of the sum of an instance of a series that fulfills the assumptions of the theorem:

**Corollary 1.** (Corollary 2 in [1]) Let \(A \in \mathbb{R}\) with \(A > 1\). Let \(\{a_n\}_{n=1}^{\infty} \in \mathbb{Z}^+,\) \(\{b_n\}_{n=1}^{\infty} \in \mathbb{Z}^+\) such that:

\[
\lim_{n \to \infty} a_n^{1/n} = A
\]

and for all sufficiently large \(n \in \mathbb{N}\) (in particular: for \(n \geq 6\))

\[
a_n^{-1} (1 + 4(2/3)^n) \leq A
\]

and

\[b_n \leq 2^{(4/3)^n-1}.\]

Then the series \(\sum_{n=1}^{\infty} \frac{b_n}{a_n}\) is an irrational number.

The above corollary is an immediate consequence of the theorem by setting \(d_n = 1 + (2/3)^n\). For the formalized proof of the corollary one more auxiliary lemma was required.
3 Irrational Rapidly Convergent Series

theory Irrationality-J-Hancl
  imports HOL - Analysis, Analysis HOL - Decision-Proc, Approximation
begin

This is the formalisation of a proof by J. Hancl, in particular of the proof
of his Theorem 3 in the paper: Irrational Rapidly Convergent Series, Rend.

The statement asserts the irrationality of the sum of a series consisting of
rational numbers defined using sequences that fulfill certain properties. Even
though the statement is number-theoretic, the proof uses only arguments
from introductory Analysis.

We show the central result (theorem Hancl3) by proof by contradiction,
by making use of some of the auxiliary lemmas. To this end, assuming
that the sum is a rational number, for a quantity ALPHA(n) we show that
ALPHA(n) ≥ 1 for all n ∈ N. After that we show that we can find an
n ∈ N for which ALPHA(n) < 1 which yields a contradiction and we thus
conclude that the sum of the series is a rational number. We finally give
an immediate application of theorem Hancl3 for a specific series (corollary
Hancl3corollary, requiring lemma summable ln plus) which corresponds to
Corollary 2 in the original paper by J. Hancl.

hide-const floatarith.Max

3.1 Misc

lemma filterlim-sequentially-iff:
  filterlim f F1 sequentially ←→ filterlim (λx. f (x+k)) F1 sequentially
unfolding filterlim-iff
  apply (subst eventually-sequentially-seg)
  by auto

lemma filterlim-realpow-sequentially-at-top:
  (x::real) > 1 =⇒ filterlim ((`^) x) at-top sequentially
  apply (rule LIMSEQ-divide-realpow-zero[THEN filterlim-inverse-at-top,of - 1, simplified])
  by auto

lemma filterlim-at-top-powr-real:
  fixes g::'b ⇒ real
  assumes filterlim f at-top F (g ----> g') F g' > 1
  shows filterlim (λx. g x powr f x) at-top F
proof –
  have filterlim (λx. ((g' + 1) / 2) powr f x) at-top F
  proof (subst filterlim-at-top-gt[of - - 1], rule+)
    fix Z assume Z > (1::real)
    have ∀ f x in F. ln Z ≤ ln (((g' + 1) / 2) powr f x)
using \texttt{assms(1)[unfolded filterlim-at-top,rule-format,of ln Z / ln ((g' + 1) / 2)]}
apply (eventually-elim)
using \langle g' > 1 \rangle by \texttt{(auto simp:ln-powr divide-simps)}
exthen show \( \forall F \ x \in F. \ Z \leq ((g' + 1) / 2) \ powr f x \)
apply (eventually-elim)
apply (subst (asm) ln-le-cancel-iff)
using \langle Z > 1, \langle g' > 1 \rangle \texttt{by auto}
qed
moreover have \( \forall F \ x \in F. \ ((g' + 1) / 2) \ powr f x \leq g x \ powr f x \)
proof
−
have \( \forall F \ x \in F. \ g x > (g'/2) \)
apply (rule order-tendstoD [OF \texttt{assms(2)}])
using \langle g' > 1 \rangle \texttt{by auto}
moreover have \( \forall F \ x \in F. \ f x > 0 \)
using \texttt{assms(1)[unfolded filterlim-at-top-dense,rule-format,of 0]}
ultimately show \?thesis
apply eventually-elim
using \langle g' > 1 \rangle \texttt{by (auto intro: powr-mono2)}
qed
ultimately show \?thesis
using \texttt{filterlim-at-top-mono} \texttt{by fast}
qed

lemma \texttt{powr-finitesum}:  
fixes \( a :: \texttt{real} \) and \( s :: \texttt{nat} \) assumes \( s \leq n \)
shows \( \prod_{j=s}^{(n::nat)}.(a powr (2^j)) = a powr (\sum_{j=s}^{n::nat}.(2^j)) \)
proof(induct \( n \))
case \( 0 \)
then show \?case by auto
next
case \( \texttt{Suc n} \)
have \?case when \( s \leq n \) using \texttt{Suc.hyps}
by (metis \texttt{Suc.prems add.commute linorder-not-le powr-add prod-nat-ivl-Suc'}
sum-cl-ivl-Suc that)
moreover have \?case when \( s = \texttt{Suc n} \)
proof−
have \( \prod_{j=s}^{\texttt{Suc n}}.a powr 2 \ ^{\ j} = (a powr 2 \ ^{\ \texttt{Suc n}}) \)
using \( s = \texttt{Suc n} \) \texttt{by simp}
also have \( \texttt{(a powr 2} ^{\ \texttt{Suc n}}) = a powr \ sum \ ((^) 2) \ \{s..Suc n\} \)
using that \texttt{by auto}
ultimately show \( \prod_{j=s}^{\texttt{Suc n}}.a powr 2 \ ^{\ j} = a powr \ sum \ ((^) 2) \ \{s..Suc n\} \)
using \( s \leq \texttt{Suc n} \) \texttt{by linarith}
qed
ultimately show \?case using \( \langle s \leq \texttt{Suc n} \rangle \texttt{by linarith} \)
qed

lemma \texttt{summable-ratio-test-tendsto}:
proof
-
-\begin{proof}
  \begin{itemize}
  \item \textbf{fixes} $f :: \text{nat} \Rightarrow 'a :: \text{banach}$
  \item \textbf{assumes} $c < 1$ \textbf{and} $\forall n. \ f \ n \neq 0$ \textbf{and} $(\lambda n. \ \text{norm} \ (\text{Suc} \ n) / \text{norm} \ (f \ n)) \\
  \quad \text{------} \to c$
  \item \textbf{shows} \textit{summable} $f$
  \end{itemize}

  \begin{proof}
    \begin{itemize}
    \item obtain $N$ \textbf{where} $\text{N-dist} \forall n \geq N. \ \text{dist} \ (\text{norm} \ (\text{Suc} \ n)) / \text{norm} \ (f \ n) \ c < \frac{1-c}{2}$
    \item using \textbf{assms} unfolding \textit{tendsto-iff} \textit{eventually-sequentially}
    \item by (meson \textit{diff-gt-0-iff-gt} \textit{zero-less-divide-iff} \textit{zero-less-numerical})
    \item \textbf{have} $\text{norm} \ (f \ (\text{Suc} \ n)) / \text{norm} \ (f \ n) \leq \frac{(1+c)}{2}$ \textbf{when} $n \geq N$ \textbf{for} $n$
    \item using $\text{N-dist}[\text{rule-format}, \text{OF} \ \text{that}] \ (c < 1)$
    \item \textbf{apply} \ (auto simp add: \textit{field-simps} \textit{dist-norm})
    \item by \textbf{argo}
    \item \textbf{then have} $\text{norm} \ (f \ (\text{Suc} \ n)) \leq \frac{(1+c)}{2} \ast \text{norm} \ (f \ n)$ \textbf{when} $n \geq N$ \textbf{for} $n$
    \item using that \textbf{assms}(2)[\text{rule-format}, \text{OF} \ \text{assms}] \textbf{by} (auto simp add: divide-simps)
    \item \textbf{moreover have} $\frac{(1+c)}{2} < 1$ using $(c < 1)$ \textbf{by} auto
    \item \textbf{ultimately show} ?thesis
    \item using $\text{summable-ratio-test}[\text{OF} - \text{N-f}]$ \textbf{by} blast
  \end{itemize}
  \end{proof}

\end{proof}

\textbf{lemma} \textit{summable-in-plus}:

\begin{proof}
  \begin{itemize}
  \item \textbf{fixes} $f :: \text{nat} \Rightarrow \text{real}$
  \item \textbf{assumes} \textit{summable} $f \ \forall n. \ f \ n > 0$
  \item \textbf{shows} \textit{summable} $(\lambda n. \ \text{ln} \ (1+f \ n))$
  \end{itemize}

  \begin{proof}
    \begin{itemize}
    \item (rule \textit{summable-comparison-test-ev}[\textit{OF} - \textbf{assms}(1)])
    \item \textbf{have} $\text{ln} \ (1+f \ n) > 0$ \textbf{for} $n$ \textbf{by} (simp add: \textbf{assms}(2) \textit{ln-gt-zero})
    \item \textbf{moreover have} $\text{ln} \ (1+f \ n) \leq f \ n$ \textbf{for} $n$
    \item \textbf{apply} \ (rule \textit{ln-add-one-self-le-self2})
    \item using \textbf{assms}(2)[\text{rule-format}, \text{OF} \ \text{assms}] \textbf{by} auto
    \item \textbf{ultimately show} \ $\forall f. \ n$ in \textit{sequentially}. \ $\text{norm} \ (\text{ln} \ (1+f \ n)) \leq f \ n$
    \item \textbf{by} (auto intro!: \textit{eventuallyI} simp add: \textit{less-imp-le})
  \end{itemize}
  \end{proof}

\end{proof}

\textbf{lemma} \textit{suminf-real-offset-le}:

\begin{proof}
  \begin{itemize}
  \item \textbf{fixes} $f :: \text{nat} \Rightarrow \text{real}$
  \item \textbf{assumes} $f \ : \ \forall i. \ 0 \leq f \ i$ \textbf{and} \textit{summable} $f$
  \item \textbf{shows} $(\sum i. \ f \ (i + k)) \leq \text{suminf} \ f$
  \end{itemize}

  \begin{proof}
    \begin{itemize}
    \item \textbf{have} $(\lambda n. \ \sum i<n. \ f \ (i + k)) \longrightarrow (\sum i. \ f \ (i + k))$
    \item using \textit{summable-sums}[\text{OF} \ (\textit{summable} \ f)]
    \item by (simp add: \textbf{assms}(2) \textit{summable-LIMSEQ} \textit{summable-ignore-initial-segment})
    \item \textbf{moreover have} $(\lambda n. \ \sum i<n. \ f \ i) \longrightarrow (\sum i. \ f \ i)$
    \item using \textit{summable-sums}[\text{OF} \ (\textit{summable} \ f)] \textbf{by} (simp add: \textit{sums-def atLeast0LessThan} \textit{f})
    \item \textbf{then have} $(\lambda n. \ \sum i<n+k. \ f \ i) \longrightarrow (\sum i. \ f \ i)$
    \item by (rule \textit{LIMSEQ-ignore-initial-segment})
    \item \textbf{ultimately show} ?thesis
    \item \textbf{proof} \ (rule \textit{LIMSEQ-le}, \textit{safe intro!}: exI[\textit{OF} - \textit{k}])
    \item \textbf{fix} $n$ \textbf{assume} $k \leq n$
  \end{itemize}
  \end{proof}

\end{proof}
have \( \sum_{i<n} f (i + k) = (\sum_{i<n} (f \circ (\lambda i. i + k)) i) \)

by simp

also have \( \ldots = (\sum_{i\in(\lambda i. i + k)^* \{..<n\}} f i) \)

by \( \text{(subst sum\_reindex) auto} \)

also have \( \ldots \leq \sum f \{..<n + k\} \)

by \( \text{(intro sum\_mono2) (auto simp: f)} \)

finally show \( \sum_{i<n} f (i + k) \leq \sum f \{..<n + k\} \).

qed

3.2 Auxiliary lemmas and the main proof

lemma factt:
fixes \( s \ n :: \text{nat} \)
assumes \( s \leq n \)
shows \( \sum_{i=s..n} 2^i < (2^{n+1}) :: \text{real} \)
using assms

proof (induct n)

next

have \( \text{case} \ 0 \)
using \( \text{that} \)
by auto

next

have \( \text{case} \ \text{when} \ s = n + 1 \)
using \( \text{that} \)
by auto

moreover have \( \text{case} \ \text{when} \ s \neq n + 1 \)

proof

define \( aa \)
where \( \prod_{j=1..n} \text{of-int}(a j) \)

finally show \( \sum_{i=s..(Suc n)} 2^i < (2^{Suc n + 1}) \)

by auto

qed

ultimately show \( \text{case} \ \text{by} \ \text{blast} \)

qed
define ff where \( \text{ff} = (\lambda i. \text{real-of-int} \ (b \ (i+1)) / \text{real-of-int} \ (a \ (i+1))) \)

have \((\sum j. \text{ff} \ (j+n)) = (\sum n. \text{ff} \ n) - \text{sum} \ \{..<n\}\)
  apply (rule suminf-minus-initial-segment)
  using assumercial

also have \(\ldots = p/q - \text{sum} \ \{..<n\}\)
  using assumercial

finally have \(\ldots \geq 1\)
  by (auto)

proof –
  have \(\sum \text{ff} \ \{..<n\} = (\sum j=1..n. \text{ff} \ (j-1))\)
    apply (subst sum-bounds-lt-plus1[symmetric])
    by simp
  then show \(?\text{thesis}\) unfolding \(\text{ff-def} \ \text{by auto}\)
qed

lemma show?:
  fixes d::nat\Rightarrow real and a b::nat\Rightarrow int and q p::int
  assumes q \geq 1 and p \geq 1 and a; \forall n. a \ n \geq 1 and \forall n. b \ n \geq 1
  and assumercial:
    \(\lambda n. \ (b \ (n+1) / a \ (n+1)) \ \text{sums} \ (p/q)\)
  shows q*((\prod j=1..n. of-int( a \ j)))*( suminf (\lambda j::nat. \ ((b \ (j+n+1) / a \ (j+n+1) ))) ) \geq 1
  (is \?L \geq -)
  proof –
    define LL where \( LL = \?L \)
    define aa where \( aa = (\prod j=1..n. \text{real-of-int} \ (a \ j)) \)
    define ff where \( \text{ff} = (\lambda i. \text{real-of-int} \ (b \ (i+1)) / \text{real-of-int} \ (a \ (i+1))) \)
  have \(?L > 0\)
  proof –
    have aa > 0
      unfolding aa-def using a
      apply (induct n, auto)
      by (simp add: int-one-le-iff-zero-less prod-pos)
    moreover have \((\sum j. \text{ff} \ (j+n)) > 0\)
    proof (rule suminf-pos)
      have \text{summable} \text{ff} unfolding \text{ff-def} using assumercial
        using \text{summable-def} by blast
      then show \text{summable} (\lambda j. \text{ff} \ (j+n)) using \text{summable-iff-shift[of \text{ff} \ n]} by auto
    show \(\forall i. 0 < \text{ff} \ (i+n)\) unfolding \text{ff-def} using a assms(4) int-one-le-iff-zero-less
    by auto
    qed
ultimately show \(?thesis\) unfolding \(aa\)-def \(ff\)-def using \((q\geq1)\) by auto

qed

moreover have \(?L \in \mathbb{Z}\)

proof –

have \(?L = aa \ast (p - q \ast (\sum j=1..n. b j / a j))\)

unfolding \(aa\)-def

apply (rule showpre7)

using assms int-one-le-iff-zero-less by auto

also have ... \(= aa \ast p - q \ast (\sum j=1..n. aa \ast b j / a j)\)

by (auto simp add: algebra-simps sum-distrib-left)

also have ... \(= \prod a \{1..n\} \ast p - q \ast (\sum j = 1..n. b j \ast \prod a \{(1..n) - \{j\}\})\)

proof –

have \((\sum j=1..n. aa \ast b j / a j) \ast (\sum j=1..n. b j \ast \prod a \{(1..n) - \{j\}\})\)

unfolding of-int-sum

proof (rule sum.cong)

fix \(x\) assume \(x \in \{1..n\}\)

then have \((\prod i = 1..n. \text{real-of-int} (a i)) = a x \ast (\prod i \in \{1..n\} - \{x\})\).

real-of-int \((a i)\)

apply (rule-tac prod.remove)

by auto

then have \(aa / \text{real-of-int} (a x) = \prod a \{(1..n) - \{x\}\}\)

unfolding \(aa\)-def using \(a[\text{rule-format},of x]\) by (auto simp add: field-simps)

then show \(aa \ast b x / a x = b x \ast \prod a \{(1..n) - \{x\}\}\)

by (metis mult.commute of-int-mult times-divide-eq-right)

qed simp

moreover have \(aa \ast p = (\prod j = 1..n. (a j)) \ast p\)

unfolding \(aa\)-def by auto

ultimately show \(?thesis\) by force

qed

also have ... \(\in \mathbb{Z}\) using Ints-of-int by blast

finally show \(?thesis\).

qed

ultimately show \(?thesis\)

apply (fold LL-def)

by (metis Ints-cases int-one-le-iff-zero-less not-less of-int-0-less-iff of-int-less-1-iff)

qed

lemma show8:

fixes \(d :: \text{nat} \Rightarrow \text{real}\) and \(a :: \text{nat} \Rightarrow \text{int}\) and \(s :: \text{nat}\)

assumes \(A > 1\) and \(d\): \(\forall n. d n > 1\) and \(a; \forall n. a n > 0\) and \(s > 0\)

and convergent-prod \(d\)

and assu2: \(\forall n \geq s. (A/\text{of-int} (a n)) \operatorname{powr}(1/\text{of-int} (2^s n)) > \operatorname{proinf} (\lambda j. \operatorname{d}(n + j))\)

shows \(\forall n \geq s. \operatorname{proinf} (\lambda j. d(j+n)) < A / (\operatorname{Max} ((\lambda j::\text{nat}). (\text{of-int} (a j)) \operatorname{powr}(1/\text{of-int} (2^s j))) \ast \{s..n\})\)

proof (rule)

fix \(n\) assume \(s \leq n\)

define \(sp\) where \(sp = (\lambda n. \operatorname{proinf} (\lambda j. d(j+n)))\)
define \texttt{ff} where \texttt{ff} = (λ(j::nat). \texttt{(real-of-int} (a \cdot j)) \texttt{powr}(1 / \texttt{of-int} (2^\cdot j)))

have \texttt{sp \ i \ ≥ \ sp \ n \ when \ i ≤ n \ for \ i}

proof —

have \((\prod j. \ d \ (j + i)) = (\prod i. \ a \cdot (i + (n - i) + i)) \cdot (\prod i. \ a < n - i \cdot \ d \ (i + i))\)

apply \(\texttt{rule prodinf-split-initial-segment}\)

subgoal using \(\texttt{convergent-prod} \ d \ \texttt{convergent-prod-iff}[[0 \ d \ i]] \ \texttt{by simp}\)

subgoal for \(j\) using \(d[[\texttt{rule-format},of} j+1]]\) \ by auto

done

then have \(\texttt{sp \ i = sp \ n \cdot (\prod j < n - i \cdot \ d \ (i + j)}\)

unfolding \(\texttt{sp-def} \ \texttt{by} \ (\texttt{auto simp:algebra-simps})\)

moreover have \(\texttt{sp \ i > 1 \ sp \ n > 1}\)

unfolding \(\texttt{sp-def} \ \texttt{using} \ \texttt{convergent-prod-iff-shift} \ \texttt{convergent-prod} \ d \ d\)

by \(\texttt{(auto intro!:less-1-prodinf)}\)

moreover have \(\(\prod j < n - 1 \cdot d \ (i + j) \geq 1\)\)

apply \(\texttt{(rule prodgc-e)}\)

using \(\texttt{d less-imp-le by auto}\)

ultimately show \(\texttt{thesis by auto}\)

qed

moreover have \(\forall \ j \geq s. \ A / ff \ j > sp \ j\)

unfolding \(\texttt{ff-def} sp-def using \texttt{assu2 by (auto simp:algebra-simps)}\)

ultimately have \(\forall \ j. s \leq j \land j \leq n \rightarrow A / ff \ j > sp \ n \ by \ texttt{force}\)

then show \(\texttt{sp \ n} < A / \texttt{Max} (ff'(\{s..\}))\)

by \(\texttt{(metis (mono-tags, hide-lams) Max-in \(\{n ≥ s\}) atLeastAtMost-iff empty-iff finite-atLeastAtMost finite-imageI imageE image-is-empty order-refl)}\)

qed

lemma auxiliary1-9:

fixes \(d ::natural\rightarrow real \ and \ a ::natural\rightarrow int \ and \ s ::natural\)

assumes \(d :: \\forall \ n. \ d \ n > 1 \ and \ a :: \\forall \ n. \ a \ n > 0 \ and \ s > 0 \ and \ n ≥ m \ and \ m ≥ s\)

and \(auxifalse-assu :: \\forall \ n ≥ m. \ (\texttt{of-int} (a \ (n+1)))) \texttt{powr}(1 / \texttt{of-int} (2^\cdot (n+1))) < (d \ (n+1)) \cdot (\texttt{Max} ((λ \ j::natural). \ (\texttt{of-int} (a \ j)) \texttt{powr}(1 / \texttt{of-int} (2^\cdot j)))) \cdot (\{s..\})\)

shows \(\texttt{(of-int} (a \ (n+1)))) \texttt{powr}(1 / \texttt{of-int} (2^\cdot (n+1))) < (\prod j=(m+1)..(n+1). \ d \ j) \cdot (\texttt{Max} ((λ \ j::natural). \ (\texttt{of-int} (a \ j)) \texttt{powr}(1 / \texttt{of-int} (2^\cdot j)))) \cdot (\{s..\})\)

proof —

define \texttt{ff} where \texttt{ff} = (λ(j::nat). \texttt{(real-of-int} (a \ j)) \texttt{powr}(1 / \texttt{of-int} (2^\cdot j)))

have \[\texttt{simp};ff \ j > 0 \ for \ j\]

unfolding \(\texttt{ff-def by (metis a less-numeral-extra(3) of-int-0-less-iff powr-gt-zero)}\)

have \(\texttt{ff-asn:ff} (n+1) < d (n+1) \cdot \texttt{Max} (ff' \{s..\}) \ \texttt{when} \ n ≥ m \ \texttt{for} \ n\)

using \(\texttt{auxifalse-assu that unfolding ff-def by simp}\)

from \(\texttt{n ≥ m}\)

have \(Q: (\texttt{Max} (ff' \{s..\})) ≤ (\prod j=(m+1)..n. \ d \ j) \cdot \texttt{Max} (ff' \{s..\})\)

proof(induct \(n\))

case 0

then show \(\texttt{case using} \ (m ≥ s) \ \texttt{by simp}\)

next
lemma (Suc n)
have \( \text{case when } m = \text{Suc } n \)
  using that by auto
moreover have \( \text{case when } m \neq \text{Suc } n \)
proof –
  have \( m \leq n \) using that Suc(2) by simp
  then have IH: \( \text{Max } (ff \cdot \{s..n\}) \leq \text{prod } d \{m + 1..n\} * \text{Max } (ff \cdot \{s..m\}) \)
    using Suc(1) by linarith
  have \( \text{Max } (ff \cdot \{s..Suc n\}) = \text{Max } (ff \cdot \{s..n\} \cup \{ff \cdot (Suc n)\}) \)
    using Suc.prems assms(5) atLeastAtMostSuc-cone by auto
  also have \( ... = \text{max } (\text{Max } (ff \cdot \{s..n\})) (ff \cdot (Suc n)) \)
    using Suc.prems assms(5) max-def sup-assoc that by auto
  also have \( ... \leq \text{max } (\text{Max } (ff \cdot \{s..n\})) (d \cdot (n+1)) * \text{Max } (ff \cdot \{s..n\}) \)
    apply (rule max_mono)
    using ff-asym[of \( n \)] (\( m \leq \text{Suc } n \)) that \( \langle \text{s \leq m} \rangle \) by auto
  also have \( ... \leq \text{Max } (ff \cdot \{s..n\}) * \text{max } 1 (d \cdot (n+1)) \)
proof –
  have \( \text{Max } (ff \cdot \{s..n\}) \geq 0 \)
    by (metis (mono-tags, hide-lams) Max-in \( \langle j. 0 < ff j : m \leq n \rangle \) assms(5)
      atLeastAtMost_iff empty_iff finite-atLeastAtMost finite-image1 imageE
      image_is_empty less_eq_real_def)
  then show \( \langle \text{thesis } \rangle \) using max-mult-distrib-right
    by (simp add: max_mult_distrib_right mult.commute)
qed
also have \( ... = \text{max } (ff \cdot \{s..n\}) \cdot d \cdot (n+1) \)
  using d[rule-format, of n+1] by auto
  also have \( ... \leq \text{prod } d \{m + 1..n\} \cdot \text{Max } (ff \cdot \{s..m\}) \cdot d \cdot (n+1) \)
  using IH d[rule-format, of n+1] by auto
  also have \( ... = \text{prod } d \{m + 1..n+1\} \cdot \text{Max } (ff \cdot \{s..m\}) \)
  using \( n \geq m \) by (simp add: prod_nat_ivl Suc' algebra_simps)
finally show \( \langle \text{thesis } \rangle \) unfolding ff-def by auto
qed
ultimately show \( \langle \text{case by simp } \rangle \)
qed
then have \( d \cdot (n+1) \cdot \text{Max } (ff \cdot \{s..n\}) \leq (\prod j=(m+1..(n+1)). d j) * (\text{Max } (ff \cdot \{s..m\})) \)
  using \( m \leq n \) d[rule-format, of Suc n] by (simp add: prod_nat_ivl Suc')
then show \( \langle \text{thesis } \rangle \) unfolding ff-asym[of \( n \)] \( \langle s \leq m \rangle \cdot \langle m \leq n \rangle \) unfolding ff-def by auto
qed

lemma show9:
  fixes \( d :: \text{nat} \Rightarrow \text{real} \) and \( a :: \text{nat} \Rightarrow \text{int} \) and \( s :: \text{nat} \) and \( A :: \text{real} \)
  assumes \( A > 1 \) and \( d :: \forall n. d \cdot n \geq 1 \) and \( a :: \forall n. a \cdot n \geq 0 \) and \( s > 0 \)
    and assu1: \( ((\lambda n. (\text{of-int } (a \cdot n))) \text{powr}(1 / \text{of-int } (2^\cdot n))) \Rightarrow A \) sequentially
    and convergent_prod d
  and \( \forall n \geq s. \text{prodinf } (\lambda j. d( n+j)) \)
    < \( A / (\text{Max } ((\lambda j :: \text{nat}. (\text{of-int } (a \cdot j))) \text{powr}(1 / \text{of-int } (2^\cdot j))) \cdot \{s..n\}) \)
shows \( \forall m \geq s. \exists n \geq m. \ (\text{of-int } (a \ (n+1))) \ \text{powr}\ (1/\text{of-int} \ (2^n (n+1))) \geq \ (d \ (n+1)) \ast (\text{Max} \ (\lambda (j::\text{nat}). \ (\text{of-int} \ (a \ j)) \ \text{powr}\ (1/\text{of-int} \ (2^j))) \cdot \{s..n\}))\)

proof (rule ccontr)

define \( ff \) where \( ff = (\lambda j::\text{nat}. \ (\text{real-of-int} \ (a \ j)) \ \text{powr}\ (1/\text{of-int} \ (2^j))) \)

assume assumptioncontra: \( \neg (\forall m \geq s. \exists n \geq m. \ (\text{of-int} \ (n+1)) \geq (d \ (n+1))) \ast (\text{Max} \ (\lambda \ \{s..n\}))\)

then obtain \( t \) where \( t \geq s \) and

\( \forall n \geq t. \ ((\text{of-int} \ (n+1)) < (d \ (n+1)) \ast (\text{Max} \ (\lambda \ \{s..n\}))\)

by fastforce

define \( B \) where \( B = \text{prodinf} \ (\lambda j. \ d(t+1+j)) \)

have \( B > 0 \) unfolding \( B \)-def

apply (rule less-0-prodinf)

subgoal using convergent-prod-iff-shift[of d t+1] (convergent-prod d)

by (auto simp: algebra-simps)

subgoal using \( d \ \text{le-less-trans} \ \text{zero-le-one} \) by blast

done

have \( A \leq B \ast \text{Max} \ (\lambda \ \{s..t\}) \)

proof (rule tendsto-le[of sequentially \( \lambda n. \ (\prod j=(t+1)..<(n+1). \ d) \ast \text{Max} \ (\lambda \ \{s..t\}) \)

\( \lambda n. \ ff \ (n+1)) \longrightarrow A \)

using assumption[folded \( ff \)-def] LIMSEQ-ignore-initial-segment by blast

have \( (\lambda n. \ \text{prod} \ d \ \{t+1..n+1\}) \longrightarrow B \)

proof –

have \( (\lambda n. \ \prod i \leq n. \ d \ (t+1+i)) \longrightarrow B \)

apply (rule convergent-prod-LIMSEQ[of \( (\lambda j. \ d(t+1+j)), \text{folded} \ \text{B-def})\])

using (convergent-prod d) convergent-prod-iff-shift[of d t+1] by (simp add: algebra-simps)

then have \( (\lambda n. \ \prod i \leq n. \ d \ (i+(t+1))) \longrightarrow B \)

using atLeast0AtMost by (auto simp: algebra-simps)

then have \( (\lambda n. \ \text{prod} \ d \ \{\{t+1\}..<n+1+t\}) \longrightarrow B \)

apply (subst (asm) prod-shift-bounds-cl-nat-ivl[symmetric])

by simp

from seq-offset-neg[OF this, of t]

show \( (\lambda n. \ \text{prod} \ d \ \{t+1..n+1\}) \longrightarrow B \)

apply (elim Lim-transform)

apply (rule LIMSEQ-I)

apply (rule exI[where \( x=t+1\)])

by auto

qed

then show \( (\lambda n. \ \text{prod} \ d \ \{t+1..n+1\}) \ast \text{Max} \ (\lambda \ \{s..t\}) \longrightarrow B \ast \text{Max} \ (\lambda \ \{s..t\}) \)

by (auto intro:tendsto-eq-intros)

have \( \forall f \ n \ \text{in sequentially.} \ (\text{of-int} \ (n+1)) < (\prod j=(t+1)..<(n+1). \ d j) \ast (\text{Max} \ (\lambda \ \{s..t\})\)

unfolding eventually-sequentially \( ff \)-def
lemma show10:
  fixes \( \ell :: \mathbb{N} \rightarrow \mathbb{R} \) and \( a :: \mathbb{N} \rightarrow \mathbb{Z} \) and \( s :: \mathbb{N} \)
  assumes \( d :: \mathbb{N} \rightarrow \mathbb{N} > 1 \) and \( e :: \mathbb{N} \rightarrow \mathbb{N} > 0 \) and \( s > 0 \)
  and \( 9 :: \forall m \geq s. \exists n \geq m. \ ((\text{of-int} (a \ (n+1))) \ 	ext{powr} (1 / \text{of-int} (2 \ ^{\ (n+1)}))) \geq \ \\ (d \ (n+1)) \ast (\text{Max} ((\lambda j :: \mathbb{N} . \ (\text{of-int} (a j))) \ 	ext{powr} (1 / \text{of-int} (2 \ ^{\ j}))) \ ^{\ \{s \cdot n\} }) )\)
  shows \( \forall m \geq s. \exists n \geq m. \ (((d \ (n+1))) \text{powr}(2 \ ^{\ (n+1)})) \ast (\prod j=1..n. \ \text{of-int}( a j)) \ast (1/(\prod j=1..s-1. \ \text{of-int}( a j) ))))) \leq (a \ (n+1)) \)
  proof (rule,rule)
    fix \( m \) assume \( s \leq m \)
    from \( 9 \mid \text{rule-format, OF this} \)
    obtain \( n \) where \( n \geq m \) and \( \text{asm-9} :: ((\text{of-int} (a \ (n+1))) \ 	ext{powr} (1 / \text{of-int} (2 \ ^{\ (n+1)}))) \geq \ \\ (d \ (n+1)) \ast (\text{Max} ((\lambda j :: \mathbb{N} . \ (\text{of-int} (a j))) \ 	ext{powr} (1 / \text{of-int} (2 \ ^{\ j}))) \ ^{\ \{s \cdot n\} }) ) \)
    by auto
    with \( s \leq m \) have \( s \leq n \) by auto
    have \( \text{prod} :: (\prod j=1..n. \ \text{real-of-int}( a j)) \ast (1/(\prod j=1..s-1. \ \text{of-int}( a j) )) = (\prod j=s..n. \ \text{of-int}( a j)) \)
    proof -
      define \( f \) where \( f = (\lambda j . \ \text{real-of-int}( a j)) \)
      have \( \{\text{Suc 0..n} \} = \{\text{Suc 0..s} = \text{Suc 0} \} \cup \{s..n\} \) using \( \langle n \geq s \rangle \ \langle s > 0 \rangle \)
      by auto
      then have \( (\prod j=1..n. \ \text{f j}) = (\prod j=1..s-1. \ \text{f j}) \ast (\prod j=s..n. \ \text{f j}) \)
apply (subst prod.union-disjoint[symmetric])
by auto
moreover have \((\prod_{j=1..s-1} f_j) > 0\)
apply (rule linordered-semidom-class.prod-pos)
using a unfolding f-def by auto
then have \((\prod_{j=1..s-1} f_j) \neq 0\) by argo
ultimately show \(\text{thesis}\) unfolding f-def by auto
qed
then have \(((d (n+1))powr(2^{\cdot(n+1)}) \cdot (\prod_{j=1..n. of-int( a j)}) \cdot (1/\prod_{j=1..s-1. (of-int( a j))})) =((d (n+1))powr(2^{\cdot(n+1)}) \cdot (\prod_{j=s..n. of-int( a j))))\)
proof –
define f where \(f = (\lambda j. \text{real-of-int}( a j))\)
define c where \(c = (d (n+1))powr(2^{\cdot(n+1)})\)
show \(\text{thesis}\) using prod
apply (fold f-def c-def)
by (metis mult.assoc)
qed
also have ...
\(\leq ((d (n+1))powr(2^{\cdot(n+1)}) \cdot (\prod_{i=s..n. (Max((\lambda j. \text{real-of-int}( a j) \ powr (1 / \text{real-of-int}( 2^{\cdot j})) ) \cdot \{s..n\} )) powr(2^{\cdot i}))))\)
proof (rule mult-left-mono)
show \(0 \leq (d (n+1)) powr 2^{\cdot (n + 1)}\) by auto
show \(\prod_{j = s..n. \text{real-of-int}( a j)} \leq (\prod_{i = s..n.} \text{Max}((\lambda j. \text{real-of-int}( a j) \ powr (1 / \text{real-of-int}( 2^{\cdot j})))) \cdot (s..n)) powr 2^{\cdot i}\)
proof (rule prod-mono)
fix i assume i: \(i \in \{s..n\}\)
have real-of-int (a i) = (real-of-int (a i) powr (1 / real-of-int (2^{\cdot i}))) powr 2^{\cdot i}
unfolding powr-powr by (simp add: a less-eq-real-def)
also have \(\ldots \leq (\text{Max}((\lambda j. \text{real-of-int}( a j) \ powr (1 / \text{real-of-int}( 2^{\cdot j})))) \cdot (s..n)) powr(2^{\cdot i})\)
proof (rule powr-mono2)
show real-of-int (a i) powr (1 / real-of-int (2^{\cdot i})) \leq \text{Max} ((\lambda j. \text{real-of-int}( a j) \ powr (1 / real-of-int (2^{\cdot j}))) \cdot (s..n))
apply (rule Max-ge)
apply auto
using i by blast
qed simp-all
finally have real-of-int (a i) \leq \text{Max} ((\lambda j. \text{real-of-int}( a j) \ powr (1 / real-of-int (2^{\cdot j}))) \cdot (s..n)) powr 2^{\cdot i}.
then show \(0 \leq \text{real-of-int}( a i) \land \text{real-of-int}( a i) \leq \text{Max} ((\lambda j. \text{real-of-int}( a j) \ powr (1 / real-of-int (2^{\cdot j}))) \cdot (s..n)) powr 2^{\cdot i}.
qed simp
\[ \{s..n\} \text{ powr } 2^{-i} \]

using \(a\) by (metis (real-of-int (a \(i\)) = (real-of-int (a \(i\)) powr (1 / real-of-int (2 ^ \(i\)))) powr 2 ^ -i ;
powr-ge-pzero)

qed

also have ... = ((d \((n+1)\)) powr (2 \(^n\)) \(\cdot\) \((s..n)\)) powr \((\sum i=s..n, 2^i)\)

proof -

have ((d \((n+1)\)) powr (2 \(^n\)) \(\ge\) 1

by (metis Transcendental.log-one \(d\) le-powr-iff zero-le-numeral zero-le-power zero-less-one)

moreover have \(\prod i=s..n, \text{Max}((\lambda j: \text{nat}. \text{of-int}(a \cdot j) powr (1 / \text{real-of-int}(2^j)) \cdot \{s..n\})) powr (2^i) = \text{Max}((\lambda j: \text{nat}. \text{of-int}(a \cdot j) powr (1 / \text{real-of-int}(2^j)) \cdot \{s..n\})) powr (\sum i=s..n, 2^i)\)

proof -

define \(ff\) where \(ff = \text{Max}((\lambda j: \text{nat}. \text{of-int}(a \cdot j) powr (1 / \text{real-of-int}(2^j)) \cdot \{s..n\})) powr (2^i)\)

show \(\text{thesis}\) apply (fold \(ff\)-def)

using \((s\le n)\) powr-finitesum by auto

qed

ultimately show \(\text{thesis}\) by auto

qed

also have ... \(\le\) ((d \((n+1)\)) powr (2 \(^n\)) \(\cdot\) \((s..n)\)) powr (\(\sum i=s..n, 2^i\))

proof -

define \(ff\) where \(ff = \text{Max}((\lambda j: \text{nat}. \text{of-int}(a \cdot j) powr (1 / \text{real-of-int}(2^j)) \cdot \{s..n\})) powr (2^i)\)

have sum ((\(\cdot\) 2) \(s..n\) \(\le\) \(2::\text{real}\) \(\cdot\) \((n + 1)\) using \text{factt} \((s\le n)\) by auto

moreover have \(I \le ff\)

proof -

define \(S\) where \(S = (\lambda j: \text{nat}. \text{of-int}(a \cdot j) powr (1 / \text{real-of-int}(2^j)) \cdot \{s..n\})\)

have finite \(S\) unfolding \(S\)-def by auto

moreover have \(S \ne\{\}\) unfolding \(S\)-def using \((s\le n)\) by auto

moreover have \(\exists x \in S, x \ge 1\)

proof -

have \((\text{of-int}(a \cdot s) powr (1 / \text{real-of-int}(2^s))) \ge 1\)

apply (rule ge-one-powr-ge-zero)

apply auto

by (simp add: a int-one-le-iff-zero-less)

moreover have \((\text{of-int}(a \cdot s) powr (1 / \text{real-of-int}(2^s))) \in S\)

unfolding \(S\)-def

apply (rule rev-image-eql[where \(x=s\)])
ultimately show \(\vdash \mathrm{thesis}\) by \texttt{auto}

\textbf{qed}

ultimately show \(\vdash \mathrm{thesis}\) by \texttt{auto}

\textbf{qed}

moreover have \(0 \leq (d \cdot (n + 1)) \cdot \text{powr} \cdot (n + 1)\) by \texttt{auto}

ultimately show \(\vdash \mathrm{thesis}\)

apply (\texttt{fold ff-def})

apply (\texttt{rule mult-left-mono})

apply (\texttt{rule powr-mono})

by \texttt{auto}

\textbf{qed}

also have \(\vdots = (\cdot \cdot \cdot ) \cdot \text{powr} \cdot (n + 1)\)

\textbf{proof}

define \(ss\) where \(ss = \lambda j. \text{real-of-int}(a_j) \cdot \text{powr}(1 / \text{real-of-int}(2^j)) \cdot \{\cdot \cdot \cdot \} \quad \{s..n\}\)

have \(d \cdot (n + 1) \geq 0\) using \(d[\text{rule-format}, (n+1)]\) by \texttt{argo}

moreover have \(\vdash \text{Max} \cdot \text{ss} \cdot \geq 0\)

\textbf{proof}

have \((a \cdot s) \cdot \text{powr}(1 / (2^s)) \geq 0\) by \texttt{auto}

moreover have \((a \cdot s) \cdot \text{powr}(1 / (2^s)) \in ss\) by unfolding \(ss\)-def

apply (\texttt{rule-tac x=s in rev-image-eqI})

using \(\langle s \leq n \rangle\) by \texttt{auto}

moreover have \(\vdash \text{finite} \cdot ss \cdot ss \neq \{\cdot \cdot \cdot \}\) by unfolding \(ss\)-def using \(\langle s \leq n \rangle\) by \texttt{auto}

ultimately show \(\vdash \mathrm{thesis}\) using \(\text{Max-ge-iff}[\text{of ss 0}]\) by \texttt{blast}

\textbf{qed}

ultimately show \(\vdash \mathrm{thesis}\)

apply (\texttt{fold ss-def})

using \(\text{powr-mult}\) by \texttt{auto}

\textbf{qed}

also have \(\vdots = (\cdot \cdot \cdot ) \cdot \text{powr} \cdot (n + 1)\)

\textbf{proof}

define \(ss\) where \(ss = \lambda j. \text{real-of-int}(a_j) \cdot \text{powr}(1 / \text{real-of-int}(2^j)) \cdot \{\cdot \cdot \cdot \} \quad \{s..n\}\)

show \(\vdash \mathrm{thesis}\)

\textbf{proof (fold ss-def, rule powr-mono2)}

have \(\text{Max} \cdot ss \cdot \geq 0\) -- NOTE: we are repeating the same proof, so it may be a good idea to put this conclusion in an outer block so that it can be reused (without reproving).

\textbf{proof}

have \((a \cdot s) \cdot \text{powr}(1 / (2^s)) \geq 0\) by \texttt{auto}

moreover have \((a \cdot s) \cdot \text{powr}(1 / (2^s)) \in ss\) by unfolding \(ss\)-def

apply (\texttt{rule-tac x=s in rev-image-eqI})

using \(\langle s \leq n \rangle\) by \texttt{auto}
moreover have finite ss ss \neq \{\} unfolding ss-def using (s \leq n) by auto
ultimately show \preceq\thesis using Max-ge iff[of ss \theta] by blast

qed

moreover have d \cdot (n + 1) \geq 0
  using d[rule-format,of n+1] by argo
ultimately show 0 \leq (d \cdot (n + 1)) \cdot Max ss by auto

show (d \cdot (n + 1)) \cdot Max ss \leq real-of-int (a \cdot (n + 1)) pour (1 / real-of-int (2 \cdot (n + 1)))
  using asm-9[folded ss-def].

qed simp

also have ... = (of-int (a \cdot (n+1)))
  by (simp add: a less-eq-real-def pos-add-strict powr-powr)
finally show \exists n \geq m. d \cdot (n + 1) powr 2 \cdot (n + 1) \cdot (\prod j = 1 \ldots n. real-of-int (a j))
  \leq real-of-int (a \cdot (n + 1)) using (n \geq m) \cdot (m \geq s)
  apply (rule-tac x=n in exf)
by auto

qed

lemma lasttoshow:

fixes a :: nat \Rightarrow real and b :: nat \Rightarrow int and q :: int and s :: nat
assumes d: \forall n. d \cdot n \geq m and a: \forall n. a \cdot n \geq m and s: \forall n. s \cdot n \leq m

and A > 1 and b: \forall n. b \cdot n \geq 1 and 9:
  \forall m \geq s. \exists n \geq m. ((of-int (a \cdot (n+1))) powr(1 / of-int (2 \cdot (n+1)))) \geq (d \cdot (n+1)) \cdot Max ((\lambda(j::nat). (of-int (a j)) powr(1 / of-int (2 \cdot j))) \cdot \{s..n\}))))

and assu3: filterlim( \lambda n. (d n) \cdot (2 \cdot n) / b n) at-top sequentially
and 5: \forall F n in sequentially. (\sum j. (b \cdot (n + j)) / (a \cdot (n + j))) \leq 2 \cdot b n / a n
shows \exists n. q \cdot ((\prod j = 1 \ldots n. real-of-int(a j))) \leq suminf (\lambda(j::nat). (b \cdot (j+n+1)) / a (j+n+1))) < 1

proof –

define as where as=(\prod j = 1 \ldots n. real-of-int (a j))

obtain n where n \geq s and n-def1: real-of-int q \geq as \cdot 2
  * real-of-int (b \cdot (n + 1)) / d \cdot (n + 1) powr 2 \cdot (n + 1) \cdot (n - 1)
  and n-def2: d \cdot (n + 1) powr 2 \cdot (n+1) \cdot (\prod j = 1 \ldots n. real-of-int (a j)) \cdot (1 / as)
  \leq real-of-int (a \cdot (n+1))
  and n-def3: (\sum j. (b \cdot (n+1+j)) / (a \cdot (n+1+j))) \leq 2 \cdot b \cdot (n+1) / a (n+1)

proof –

have *:(\lambda n. real-of-int (b n) / d n \cdot 2 \cdot n) \longrightarrow 0
  using tendsto-inverse-0-at-top[OF assu3] by auto

then have (\lambda n. real-of-int (b n) / d n powr 2 \cdot n) \longrightarrow 0

proof –

have d n \cdot 2 \cdot n = d n powr (of-nat (2 \cdot n)) for n
  apply ( subst powr-realpow)
  using d[rule-format, of n] by auto

then show \preceq\thesis using * by auto
qed
from tendsto-mult-right-zero[OF this, OF q * as * 2]
have \((\lambda n. q * as * 2 * b n / d n powr 2 ^ n) \longrightarrow 0\)
  by auto
then have \(\forall F. n \in sequentially. q * as * 2 * b n / d n powr 2 ^ n < 1\)
  by (elim order-tendstoD) simp
then have \(\forall F. n \in sequentially. q * as * 2 * b n / d n powr 2 ^ n < 1\)
  \(\land (\sum j. (b (n + j)) / (a (n + j))) \leq 2 * b n / a n\)
  using 5 by eventually-elim auto
then obtain \(N \) where \(N-def:\forall n \geq N. q * as * 2 * b n / d n powr 2 ^ n < 1\)
  \(\land (\sum j. (b (n + j)) / (a (n + j))) \leq 2 * b n / a n\)
  unfolding eventually-sequentially by auto
obtain \(n \) where \(n-def: n \geq s \land n \geq N \land n-def:d (n + 1) powr 2 ^ (n + 1)\)
  \(\ast (\prod j = 1..n. of-int (a j)) * (1 / as) \leq \text{real-of-int} (a (n + 1))\)
  using show10[OF d a s>0] 9.folded as-def, rule-format, of max s N] by auto
with \(N-def\text{-rule-format, of n+1} \) that[of n] show \(?thesis\) by auto

qed

define \(pa\) where \(pa = (\prod j = 1..n. \text{real-of-int} (a j))\)
define \(dn\) where \(dn = d (n + 1) powr 2 ^ (n + 1)\)
have [simp]: \(dn > 0\) as \(> 0\)

subgoal unfolding \(dn-def\) by (metis d not-le numeral-One powr-gt-zero zero-le-numeral)
subgoal unfolding as-def by (simp add: a prod-pos)
done

have [simp]: \(pa > 0\)
  unfolding \(pa-def\) using \(a\) by (simp add: prod-pos)

have \(K: q * (\prod j=1..n. \text{real-of-int} (a j)) * \text{suminf} \ (\lambda (j::nat). (b (j+n+1) / a (j+n+1)))\)
  \(\leq q * (\prod j=1..n. \text{real-of-int} (a j)) * 2 * (b (n+1)) / (a (n+1))\)
  apply (fold \(pa-def\))
  using mult-left-mono[OF \(n-def3\), of real-of-int q * pa]
  \(\langle n \geq s ; \langle pa > 0 \rangle ; \langle q > 0 \rangle \) by (auto simp add: algebra-simps)
also have \(KK: \leq 2 * q * (\prod j=1..n. \text{real-of-int} (a j)) * (b (n+1)) *\)
  \((\prod j=1..n-1. \text{real-of-int} (a j)) / ((d (n+1)) powr (2 ^ (n+1))) * (\prod j=1..n. \text{real-of-int} (a j)))\)
proof -
  have \(dn * pa * (1 / as) \leq \text{real-of-int} (a (n + 1))\)
    using \(n-def2\) unfolding \(dn-def\) \(pa-def\).
  then show \(?thesis\)
    using \(pa > 0\) \(\langle q > 0 \rangle \) a\[rule-format, of Suc n] b[rule-format, of Suc n]
    by (auto simp add: divide-simps algebra-simps)
qed
also have \(KK: \ast = (q * (\prod j=1..(s-1). \text{real-of-int}(a j))) * 2\)
  \(\ast (b (n+1)) / ((d (n+1)) powr (2 ^ (n+1)))\)
  apply (fold as-def \(pa-def\) \(dn-def\))
  apply simp
  using \(0 < pa\) by blast
also have \( KKKK: \ldots < 1 \) using \( n \text{-def}1 \) unfolding as-def by simp
finally show ?thesis by auto
qed

lemma
fixes \( d : \text{n} \Rightarrow \text{real} \) and \( a \cdot b : \text{n} \Rightarrow \text{int} \) and \( A : \text{real} \)
assumes \( A > 1 \) and \( d : \forall \cdot n \cdot d \cdot n > 1 \) and \( a: \forall \cdot n \cdot a \cdot n > 0 \) and \( b: \forall \cdot n \cdot b \cdot n > 0 \)
and assu1: \( (\lambda \cdot n \cdot (\text{of-int} \cdot (a \cdot n)) \cdot \text{power} \cdot (1 \cdot \text{of-int} \cdot (2^{\cdot n}))) \longrightarrow A \) sequentially
and assu3: filterlim \( (\lambda \cdot n \cdot (d \cdot n) \cdot (2^{\cdot n})/ b \cdot n) \) at-top sequentially
and convergent-prod \( d \)
shows issummable: summable \((\lambda \cdot j \cdot b \cdot j / a \cdot j)\)
and show5: \( \forall \cdot F \cdot n \cdot \text{in sequentially}. (\sum j \cdot (b \cdot (n + j)) / (a \cdot (n + j))) \leq 2 * b \cdot n / a \cdot n \)

proof –
define \( c \) where \( c = (\lambda \cdot j \cdot b \cdot j / a \cdot j) \)
have \( c\cdot\text{pos}\cdot c \cdot j > 0 \) for \( j \)
using \( a \cdot b \) unfolding c-def by simp
have \( c\cdot\text{ratio}\cdot\text{tendsto} : (\lambda \cdot n \cdot (c \cdot (n + 1)) / c \cdot n) \longrightarrow 0 \)
proof –
define \( nn \) where \( nn = (\lambda \cdot n \cdot (2::\text{int})^{\cdot} (\text{Suc} \cdot n)) \)
define \( ff \) where \( ff = (\lambda \cdot n \cdot (a \cdot n / a \cdot (\text{Suc} \cdot n)) \cdot \text{power} \cdot (1 / \text{nn} \cdot n) \cdot \text{of}(d \cdot (\text{Suc} \cdot n))) \)
have \( \text{nn}\cdot\text{pos}\cdot\text{nn} \cdot n > 0 \) and \( \text{ff}\cdot\text{pos}\cdot\text{ff} \cdot n > 0 \) for \( n \)
subgoal unfolding \( \text{nn}\cdot\text{def} \) by simp
subgoal unfolding \( \text{ff}\cdot\text{def} \)
using \( d\cdot\text{rule}\cdot\text{f:\text{format}} \cdot \text{of} \cdot \text{Suc} \cdot n \cdot a\cdot\text{rule}\cdot\text{format} \cdot \text{of} \cdot n \cdot a\cdot\text{rule}\cdot\text{format} \cdot \text{of} \cdot \text{Suc} \cdot n \)
by auto
done
have \( \text{ff}\cdot\text{tendsto}\cdot\text{ff} \longrightarrow 1 / \text{sqrt} \cdot A \)
proof –
have \( (\text{of-int} \cdot (a \cdot n)) \cdot \text{power} \cdot (1 / (\text{nn} \cdot n)) = \text{sqrt}(\text{of-int} \cdot (a \cdot n)) \cdot \text{power} \cdot (1 / \text{of-int} \cdot (2^{\cdot n}))) \) for \( n \)
unfolding \( \text{nn}\cdot\text{def} \) using \( a \)
apply \( \text{simp add: power-half-sqrt \cdot \text{symmetric}} \)
by \( \text{simp add: power-half-sqrt \cdot \text{symmetric}} \cdot \text{linordered-field-class.sign-simps}(2^{\cdot}1) \cdot \text{power-power} \)
moreover have \( (\lambda \cdot n \cdot \text{sqrt}(\text{of-int} \cdot (a \cdot n)) \cdot \text{power} \cdot (1 / \text{of-int} \cdot (2^{\cdot n}))) \longrightarrow \text{sqrt} \cdot A \) sequentially
using \( \text{assu1} \) tendsto-real-sqrt by blast
ultimately have \( (\lambda \cdot n \cdot (\text{of-int} \cdot (a \cdot n)) \cdot \text{power} \cdot (1 / \text{of-int} \cdot (\text{nn} \cdot n))) \longrightarrow \text{sqrt} \cdot A \)
sequentially
by auto
from tendsto-divide[\text{OF this assu1}[\text{THEN} \text{LIMSEQ}\cdot\text{ignore-initial-segment\cdot\text{where} \( k=1 \)]]
have \( \text{(\lambda} \cdot n \cdot (a \cdot n / a \cdot (\text{Suc} \cdot n)) \cdot \text{power} \cdot (1 / \text{nn} \cdot n)) \longrightarrow 1 / \text{sqrt} \cdot A \)
using \( \langle A > 1 \rangle \cdot a \) unfolding \( \text{nn}\cdot\text{def} \)
by \( \text{auto simp add: power-divide inverse-eq-divide sqrt-divide-self-eq} \)
moreover have \( \langle \lambda \cdot n \cdot d \cdot (\text{Suc} \cdot n) \rangle \longrightarrow 1 \)
apply \( \text{rule convergent-prod-imp-LIMSEQ} \)
using convergent-prod-iff-shift[\text{of} \cdot d \cdot 1] \cdot \text{convergent-prod} \cdot d \) by auto
ultimately show \( ?thesis \)
unfolding \( ff\)-def by (auto intro:tendsto-eq-intros)

qed

have \( (\lambda n. (ff\ n)\ powr\ nn\ n) \longrightarrow 0 \)
proof –
define \( aa \) where \( aa = (1 + 1/\sqrt{A}) / 2 \)
have eventually \( (\lambda n. ff\ n < aa) \) sequentially
  apply (rule order-tendstoD[OF ff-tendsto])
unfolding \( aa\)-def using \( \langle A > 1 \rangle \) by (auto simp add:field-simps)
moreover have \( (\lambda n. aa powr\ nn\ n) \longrightarrow 0 \)
proof –
have \( (\lambda y. aa ^ (nat\ ◦\ nn)\ y) \longrightarrow 0 \)
  apply (rule tendsto-power-zero)
  subgoal unfolding \( nn\)-def comp-def
    apply (rule filterlim-subseq)
    by (auto intro:strict-monoI)
  subgoal unfolding \( aa\)-def using \( \langle A > 1 \rangle \) by auto
  done
then show \( ?thesis \)
proof (elim filterlim-mono-eventually)
  have \( aa > 0 \) unfolding \( aa\)-def using \( \langle A > 1 \rangle \)
    by (auto simp add:field-simps pos-add-strict)
then show \( \forall F\ x\ in\ sequentially.\ aa ^ (nat\ ◦\ nn)\ x = aa powr\ real-of-int \nn\ x \)
  by (auto simp:powr-int order.strict-implies-order[OF nn-pos])
qed auto

ultimately show \( ?thesis \)
apply (elim metric-tendsto-imp-tendsto)
apply (auto intro:powr-mono2 elim:eventually-mono)
using \( nn\)-pos \( ff\)-pos
proof (elim filterlim-mono-eventually)
  show \( \forall F\ x\ in\ sequentially.\ ff\ x\ powr\ (nn\ x) = d\ (Suc\ x) ^ (nat\ (nn\ x)) * (a\ x / a\ (Suc\ x)) \)
    apply (rule eventuallyI)
    subgoal for \( x \)
      unfolding \( ff\)-def
      using \( a[rule-format,of\ x]\ a[rule-format,of\ Suc\ x]\ d[rule-format,of\ Suc\ x] \nn\)-pos[of \( x \)]
      apply (auto simp add:field-simps powr-divide powr-powr powr-mult)
      by (simp add:powr-int)
    done
qed auto

moreover have \( (\lambda n. b\ (Suc\ n) / (d\ (Suc\ n)) ^ (nat\ (nn\ n))) \longrightarrow 0 \)
using tendsto-inverse-0-at-top[OF assu3,THEN LIMSEQ-ignore-initial-segment[where \( k = 1 \)]
unfolding \( nn\)-def by (auto simp add:field-simps nat-mult-distrib nat-power-eq)
ultimately have \((\lambda n. b (\text{Suc } n) \cdot (a n / a (\text{Suc } n))) \longrightarrow 0\)

apply

subgoal premises \(asm\)
  using tendsto-mult[OF \(asm\),simplified]
  apply (elim filterlim-mono-eventually)
using \(d\) by (auto simp add: algebra-simps,metis (mono-tags, lifting) always-eventually not-one-less-zero)
done
then have \((\lambda n. (b (\text{Suc } n) / b n) \cdot (a n / a (\text{Suc } n))) \longrightarrow 0\)
apply (elim Lim-transform-bound[rotated])
apply (rule eventuallyI)
subgoal for \(x\) using \(a[\text{rule-format}, \text{of } x] a[\text{rule-format}, \text{of } \text{Suc } x]\)
  \(b[\text{rule-format}, \text{of } x] b[\text{rule-format}, \text{of } \text{Suc } x]\)
by (auto simp add: algebra-simps)
done
then show \(?thesis\) unfolding \(c\)-def by (auto simp add: algebra-simps)
qed

from \(c\)-ratio-tendsto
have \((\lambda n. \text{norm} (b (\text{Suc } n) / a (\text{Suc } n)) / \text{norm} (b n / a n)) \longrightarrow 0\)

unfolding \(c\)-def
apply (elim Lim-transform-bound[rotated])
apply (rule eventuallyI)
using \(a b\) by (auto simp add: divide-simps abs-of-pos)

from summable-ratio-test-tendsto[OF - - this \(a b\)]
show summable \(c\) unfolding \(c\)-def by auto

by (metis less-irrefl)

have \(\forall F \ n \in \text{sequentially}. (\sum j. c (n + j)) \leq 2 \cdot c\ n\)

proof –
  obtain \(N\) where \(N\)-ratio: \(\forall n \geq N\). \(c (n + 1) / c n < 1 / 2\)
  proof –
    have eventually \((\lambda n. c (n+1) / c n < 1/2)\) sequentially
      using \(c\)-ratio-tendsto[unfolded tendsto-iff,rule-format, of \(1/2\),simplified]
      apply eventually-elim
      subgoal for \(n\) using \(c\)-pos[of \(n\)] \(c\)-pos[of \(\text{Suc } n\)] by auto
done
then show \(?thesis\) using that unfolding eventually-sequentially by auto
done

have \((\sum j. c (j + n)) \leq 2 \cdot c\ n\ \text{when} \ n \geq N \ \text{for} \ n\)

proof –
  have \((\sum j<\text{m}. c (j + n)) \leq 2\cdot c\ n \cdot (1 - 1 / 2^{\text{m}(\text{m}+1)})\) for \(m\)
  proof (induct \(m\))
    case 0
    then show \(?case\) using \(c\)-pos[of \(n\)] by simp
  next
    case (Suc \(m\))
    have \((\sum j<\text{Suc } m. c (j + n)) = c\ n + (\sum i<\text{m}. c (\text{Suc } i + n))\)
      unfolding sum-lessThan-Suc-shift by simp
also have \( \cdots \leq c \cdot n + (\sum i < m. c (i + n) / 2) \)

**proof**

have \( c (\text{Suc} \ i + n) \leq c (i + n) / 2 \) for \( i \)

using \( N\)-ratio[rule-format,of \( i+n \)] \( \langle \forall n \geq N \rangle \) \( c\)-pos[of \( i+n \)] by \( \text{simp} \)

then show \( \text{thesis} \) by (auto intro:sum-mono)

**qed**

also have ... = \( c \cdot n + (\sum i < m. c (i + n)) / 2 \)

unfolding \( \text{sum-divide-distrib} \) by \( \text{simp} \)

also have ... \( \leq c \cdot n + c \cdot n \cdot \sum (1 - 1 / 2 \cdot (m + 1)) \)

using \( \text{Suc} \) by \( \text{auto} \)

also have ... = \( 2 \cdot c \cdot n \cdot \sum (1 - 1 / 2 \cdot (\text{Suc} \ m + 1)) \)

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

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

**qed**

then have \( (\sum j < m. c (j + n)) \leq 2 \cdot c \cdot n \) for \( m \)

using \( c\)-pos[of \( n \)]

by (smt divide-le-eq-1-pos divide-pos-pos nonzero-mult-div-cancel-left zero-less-power)

moreover have \( \text{summable} \ (\lambda j. c (j + n)) \)

using \( \langle \text{summable} \rangle \) by (simp add: summable-iff-shift)

ultimately show \( \text{thesis} \) using \( \text{suminf-le-const[of \( \lambda j. c (j+n) \cdot 2 \cdot c \cdot n \]} \) by \( \text{auto} \)

**qed**

then show \( \text{thesis unfolding eventually-sequentially} \) by (auto \( \text{simp add:field-simps} \))

**qed**

then show \( \forall \langle f \rangle n \in \text{sequentially}. \ (\sum j. (b (n + j)) / (a (n + j))) \leq 2 \cdot b \cdot n / a \cdot n \)

unfolding \( c\)-def by \( \text{simp} \)

**qed**

**theorem** \( \text{Hancl3} \):

*fixes* \( d :: \text{nat}\Rightarrow\text{real} \) and \( a \ b :: \text{nat}\Rightarrow\text{int} \)

*assumes* \( A > 1 \) and \( d;\forall n. \ d \cdot n > 1 \) and \( a;\forall n. \ a \cdot n > 0 \) and \( b;\forall n. \ b \cdot n > 0 \) and \( s > 0 \)

\( \text{and} \ \text{assu1: } (\lambda n. \ (\text{of-int} \ a \cdot n)) \ \text{powr} (1 / (\text{of-int} \ (2 \cdot n))) \rightarrow A \) sequentially

\( \text{and} \ \text{assu2: } \forall n \geq s. \ (A/ \ (\text{of-int} \ a \cdot n)) \text{powr} (1 / (\text{of-int} \ (2 \cdot n))) > \text{prodinf} \ (\lambda j. d(n + j)) \)

\( \text{and} \ \text{assu3: } \text{filterlim} \ (\lambda n. \ (d \cdot n) ^ (2 \cdot n) / b \cdot n) \text{ at-top sequentially} \)

\( \text{and} \ \text{convergent-prod d} \)

*shows* \( \text{suminf} (\lambda n. \ b \cdot n / a \cdot n) \notin \text{Rats} \)

**proof** (rule \( \text{ccontr} \))

*assume* \( \text{asm:= } (\text{suminf} (\lambda n. \ b \cdot n / a \cdot n) \notin \text{Rats}) \)

*have* \( \text{ab-sum:summable} \ (\lambda j. \ b \cdot j / a \cdot j) \)

*using* \( \text{issummable[ OF } \text{A>1, d a b \ text{assu1 \ text{assu3 \ convergent-prod d }]} \).

*obtain* \( p q :: \text{int} \) where \( q > 0 \) and \( \text{pq-def:( } (\lambda n. \ b \cdot (n+1) / a \cdot (n+1)) \ ) \text{ sums (p/q)} \)

**proof**

*from* \( \text{asm} \) *have* \( \text{suminf} (\lambda n. \ b \cdot n / a \cdot n) \in \text{Rats} \) by \( \text{auto} \)

*then have* \( \text{suminf} (\lambda n. \ b \cdot (n+1) / a \cdot (n+1)) \in \text{Rats} \)

*apply* (subst \( \text{suminf-minus-initial-segment[OF ab-sum,of 1]} \))

21
define ALPHA where

\[\text{ALPHA} = (\lambda n. (\text{of-int } q)\cdot (\Pi j=1\ldots n. \text{of-int}(a j))\cdot (\suminf (\lambda (j::\text{nat}). (b (j+n+1)/a (j+n+1))))\]

have \(\text{ALPHA } n \geq 1\) for \(n\)

proof

- have \(\suminf (\lambda n. b (n+1)/a (n+1)) > 0\)
  apply (rule suminf-pos)
  subgoal using summable-ignore-initial-segment[of ab-sum, of 1] by auto
  subgoal using \(a\) by simp
done

then have \(p/q > 0\) using sums-unique[of pq-def, symmetric] by auto

then have \(q\geq 1\) \(p\geq 1\) using \(\langle q > 0\rangle\) by (auto simp add: divide-simps)

moreover have \(\forall n. I \leq a n \forall n. I \leq b n\) using \(a\) \(b\)
  by (auto simp add: int-one-iff-zero-less)

ultimately show \(?thesis\) unfolding ALPHA-def
  using show7[of \(\langle OF - - - - pq-def\rangle\)] by auto
qed

moreover have \(\exists n. \text{ALPHA } n < 1\) unfolding ALPHA-def

proof (rule lasttoshow[of \(\langle OF d a (s>0) \langle q>0\rangle \langle A>1\rangle b - \text{assu3}\)]

  show \(\forall F\) \(n\) in sequentially. \((\sum j. \text{real-of-int } (b (n + j)) / \text{real-of-int } (a (n + j)))\)
    \(\leq\) \(\text{real-of-int } (2 \cdot b^n) / \text{real-of-int } (a n)\)
  using show5[of \(\langle OF \langle A>1\rangle d a \text{ assu1 assu3 \langle convergent-prod d \rangle}\)] by simp
  show \(\forall m\geq s. \exists n\geq m. d (n + 1) * \text{Max } ((\lambda j. \text{real-of-int } (a j) \text{ powr } (1 / \text{real-of-int } (2 \cdot j))) \cdot \{s\ldots n\})\)
    \(\leq\) \(\text{real-of-int } (a (n + 1)) \text{ powr } (1 / \text{real-of-int } (2 \cdot (n + 1)))\)
  apply (rule show9[of \(\langle OF \langle A>1\rangle d a (s>0) \text{ assu1 \langle convergent-prod d \rangle}\)]
    using show8[of \(\langle OF \langle A>1\rangle d a (s>0) \text{ convergent-prod d assu2}\)] by (simp add: algebra-simps)
qed

ultimately show False using not-le by blast
qed

corollary Hancl3corollary:

fixes \(A\) :: real and \(a\) : nat => int
assumes \(A \geq 1\) and \(a\) : \(\forall n. a n > 0\) and \(b\) : \(\forall n. b n > 0\)
and \(\text{assu1: } ((\lambda n. (\text{of-int } (a n)) \text{ powr } (1 / \text{of-int } (2^n))) \longrightarrow A)\) sequentially
and \texttt{asscor2: }\forall n \geq 6. (\texttt{af-int (a n) powr(1 /of-int (2^\circ n))})*(1 + 4*(2/3) ^n) \\
\leq A \land (b n \leq 2 \texttt{ powr((4/3) ^{(n-1)})} ) \\
\text{shows }\texttt{suminf(\lambda n. b n / a n ) \notin Rats}

\textbf{proof –}

\texttt{define d::nat} ⇒ \texttt{real where d= (\lambda n. 1+(2/3) ^{(n+1)})}

\texttt{have dgt1:\forall n. 1 < d n }\texttt{ unfolding d-def by auto}

\texttt{moreover have convergent-prod d unfolding d-def}

\texttt{apply (rule abs-convergent-prod-imp-convergent-prod)}

\texttt{apply (rule summable-imp-abs-convergent-prod)}

\texttt{using summable-ignore-initial-segment[of complete-algebra-summable-geometric [of 2/3::real,simplified],of 1] by simp}

\texttt{moreover have \forall n\geq 6. (\prod j. d (n+j)) < A / real-of-int (a n) powr (1 / real-of-int (2 \circ n))}

\texttt{proof rule+}

\texttt{fix n::nat assume 6 \leq n}

\texttt{have d-sum:summable (\lambda j. ln (d j)) unfolding d-def}

\texttt{apply (rule summable-ln-plus)}

\texttt{apply (rule summable-ignore-initial-segment[of complete-algebra-summable-geometric [of 2/3::real,simplified],of 1])}

\texttt{by simp}

\texttt{have (\sum j. ln (d (n+j))) < ln (1+4 * (2 / 3) ^ n)}

\texttt{proof –}

\texttt{define c::real where c=(2 / 3) ^ n}

\texttt{have 0 < c c < 1/8}

\texttt{proof –}

\texttt{have c=(2 / 3) ^6 * (2 / 3) ^ (n-6) unfolding c-def using (n\geq 6)}

\texttt{using (subst power-add[symmetric])}

\texttt{by auto}

\texttt{also have ... \leq (2 / 3) ^6 by (auto intro:power-le-one)}

\texttt{also have ... < 1/8 by (auto simp add:field-simps)}

\texttt{finally show c < 1/8}.

\texttt{qed (simp add:c-def)}

\texttt{have (\sum j. ln (d (n+j))) \leq (\sum j. (2 / 3) ^ (n + j + 1))}

\texttt{apply (rule suminf-le)}

\texttt{subgoal unfolding d-def}

\texttt{apply (intro allI ln-add-one-self-le-self2 )}

\texttt{apply (rule order.strict-trans[of - 0])}

\texttt{by auto}

\texttt{subgoal using summable-ignore-initial-segment[of d-sum,of n]}

\texttt{by (auto simp add:algbera-simps)}

\texttt{subgoal using summable-geometric[THEN summable-ignore-initial-segment,of 2/3 n+1]}

\texttt{by (auto simp add:algbera-simps)}

23
done
also have ... = (∑j. (2 / 3)ˆ(n+1))*(2 / 3)ˆ j
  by (auto simp add: algebra-simps power-add)
also have ... = (2 / 3)ˆ(n+1) * (∑j. (2 / 3)ˆ j)
  apply (rule suminf-mult)
  by (auto intro: summable-geometric)
also have ... = 2 * c

unfolding c-def
apply (subst suminf-geometric)
by auto
also have ... < 4 * c - (4 * c)²
using ⟨0 < c : c < 1/8⟩
by (sos ((([(A<0 * A<1) * R<1] + [(A<=0 * R<1) * (R<1/16 * [1]²)])))
also have ... ≤ ln (1 + 4 * c)
  apply (rule ln-one-plus-pos-lower-bound)
  using ⟨0 < c : c < 1/8⟩ by auto
finally show ?thesis unfolding c-def by simp
qed

then have exp (∑j. ln (d (n + j))) < 1 + 4 * (2 / 3)ˆ n
by (metis Groups.mult-ac(2) add.right-neutral add-mono-thms-linordered-field(5))

divide-inverse divide-less-eq-numeralI(1) divide-pos-pos exp-gt-zero less-eq-real-def
ln-exp ln-less-cancel-iff mult-zero-left rel-simps(27) rel-simps(76) zero-less-one
zero-less-power

moreover have exp (∑j. ln (d (n + j))) = (∏j. d (n + j))
proof (subst exp-suminf-prodinf-real [symmetric])

show ∃k. 0 ≤ ln (d (n + k))
  using dgt1 by (simp add: less-imp-le)
show abs-convergent-prod (λna. exp (ln (d (n + na))))
  (subst exp-ln)
subgoal for j using dgt1[rule-format, of n+j] by auto
subgoal unfolding abs-convergent-prod-def real-norm-def
  apply (subst abs-of-nonneg)
    using convergent-prod-iff-shift[of d n] (convergent-prod d)
    by (auto simp add: dgt1 less-imp-le algebra-simps)
done

show (∏na. exp (ln (d (n + na)))) = (∏j. d (n + j))
  apply (subst exp-ln)
subgoal using dgt1 le-less-trans zero-le-one by blast
subgoal by simp
done

qed

ultimately have (∏j. d (n + j)) < 1 + 4 * (2 / 3)ˆ n
by simp
also have ... ≤ A / (a n) powr (1 / of-int (2 ^ n))
proof —
  have a n powr (1 / real-of-int (2 ^ n)) > 0
    using a[rule-format, of n] by auto
then show ?thesis using asscor2[rule-format,OF 6≤n]
by (auto simp add:field-simps)
qed

finally show \( \prod j. d \ (n \ + \ j) < A / \ \text{real-of-int} \ (a \ n) \ \text{powr} \ (1 / \ \text{of-int} \ (2 \ ^ n)) \) .

qed

moreover have \( \lim n \ \text{sequentially.} \ d \ n \ ^ 2 \ ^ n \ / \ \text{real-of-int} \ (b \ n) :> \ \text{at-top} \)

proof

have \( \lim n \ \text{sequentially.} \ d \ n \ ^ 2 \ ^ n \ / \ 2 \ \text{powr}((4/3) \ ^ *(n-1)) :> \ \text{at-top} \)

proof

define \( n1 \) where \( n1 = (\lambda n. \ (2::real) \ ^ *(3/2) \ ^ *(n-1)) \)

define \( n2 \) where \( n2 = (\lambda n. \ ((4::real) / 3) \ ^ *(n-1)) \)

have \( \lim n \ \text{sequentially.} \ (((1+(8/9)/(n1 n)) \ \text{powr} \ (n1 n))/2) \ \text{powr} \ (n2 n) :> \ \text{at-top} \)

proof (rule filterlim-at-top-powr [where \( g' = \exp (8/9) / (2::real) \])

define \( e1 \) where \( e1 = \exp (8/9) / (2::real) \)

show \( e1 > 1 \) unfolding \( e1 \)-def by (approximation 4)

show \( \lambda n. \ ((1+(8/9))/(n1 n)) \ \text{powr} \ (n1 n))/2 \) ----> \( e1 \)

proof

have \( \lambda n. \ ((1+(8/9))/(n1 n)) \ \text{powr} \ (n1 n) \) ----> \( \exp (8/9) \)

apply (rule filterlim-compose [OF \( \text{tendsto-exp-limit-at-top} \)])

unfolding \( n1 \)-def

by (auto intro: filterlim-tendsto-pos-mult-at-top

filterlim-realpow-sequentially-at-top

simp: filterlim-sequentially-iff [of \( \lambda x. \ (3 / 2) \ ^ (x - \ Suc 0) - 1) \])

then show \( \text{thesis} \) unfolding \( e1 \)-def

by (intro \( \text{tendsto-intros,auto} \))

qed

show filterlim \( n2 \) \( \text{at-top} \) \( \text{sequentially} \)

unfolding \( n2 \)-def

apply (simp_all filterlim-sequentially-iff [of \( \lambda n. \ (4 / 3) \ ^ *(n - 1) - 1) \])

by (auto intro: filterlim-realpow-sequentially-at-top)

qed

moreover have \( \forall x \ n \ \text{in} \ \text{sequentially.} \ (((1+(8/9))/(n1 n)) \ \text{powr} \ (n1 n))/2 \)

\( \text{powr} \ (n2 n) \)

\( = d \ n \ ^ 2 \ ^ n / 2 \ \text{powr}((4/3) \ ^ *(n-1)) \)

proof (rule eventually-sequentially [of 1])

fix \( x \) assume \( x \geq (1::nat) \)

have \( (((1 + 8 / 9 / n1 x) \ \text{powr} \ n1 x / 2) \ \text{powr} \ n2 x \)

\( = (((1 + 8 / 9 / n1 x) \ \text{powr} \ n1 x) \ \text{powr} \ n2 x) / 2 \ \text{powr} \ (4 / 3) \)

\( ^ \ (x - 1) \)

apply (subst \( \text{powr-divide} \))

apply (simp_all add: \( n1 \)-def \( n2 \)-def)

by (smt \( \text{divide-nonneg-nonneg \ zero-\le-power} \))

also have \( \ldots = (1 + 8 / 9 / n1 x) \ \text{powr} \ (n1 x * n2 x) / 2 \ \text{powr} \ (4 / 3) \)^ \( x - 1 \)

apply (subst \( \text{powr-powr} \))

by simp

also have \( \ldots = (1 + 8 / 9 / n1 x) \ \text{powr} \ (2 \ ^ x) / 2 \ \text{powr} \ (4 / 3) \ ^*(x - 1) \)

25
proof
  have \( \text{n1} \times \text{n2} \times 2 \times x \)
  unfolding \text{n1-def} \text{n2-def}
  apply (subst mult.assoc)
  apply (subst power-mult-distrib[symmetric])
  using \( x \geq 1 \) by (auto simp add: power-Suc[symmetric] simp del:power-Suc)
  then show \( \text{thesis} \) by simp
qed
also have \( \ldots = (1 + 8 / 9 / \text{n1} \times) \times (2 / x) / 2 \text{ powr} (4 / 3) \times (x - 1) \)
  apply (subst (3) power-realpow[symmetric])
  apply (simp-all add: \text{n1-def})
  by (smt zero-le-divide-iff zero-le-power)
also have \( \ldots = d \times 2 \times x / 2 \text{ powr} (4 / 3) \times (x - 1) \)
proof
  define \( x_1 \) where \( x_1 = x - 1 \)
  have 
  unfolding \text{x1-def} using \( x \geq 1 \) by simp
  have 
  unfolding \text{n1-def} using \( x \geq 1 \)
  apply (fold \text{x1-def} ![symmetric])
  by (auto simp add: divide-simps)
  then show \( \text{thesis} \) unfolding \text{d-def}
  apply (subst ![symmetric])
  by auto
  qed
finally show \( ((1 + 8 / 9 / \text{n1} \times) \text{ powr} \text{n1} \times / 2) \text{ powr} n2 \times \)
  \( = d \times 2 \times x / 2 \text{ powr} (4 / 3) \times (x - 1) \).
  qed
ultimately show \( \text{thesis} \) using filterlim-cong by fast
qed
moreover have \( \forall F. n \text{ in sequentially. } d \times 2 \times n / 2 \text{ powr}((4/3)\times(n-1)) \)
  \( \leq d \times n \times n / \text{ real-of-int} (b \ n) \)
  apply (rule eventually-sequentiallyI[of 6])
  apply (rule divide-left-mono)
  subgoal for \( x \)
  using asscor2[rule-format,of \( x \)] by auto
  subgoal for \( x \)
  using \( \forall n. 1 < d \times n \text{ rule-format, of } x \) by auto
  subgoal for \( x \)
  using \( b \) by auto
  done
ultimately show \( \text{thesis} \) by (auto elim: filterlim-at-top-mono)
qed
ultimately show \( \text{thesis} \) using Hancl2[OF \( A > 1 \) - \( a b - \text{assu1}, of d \ 6 \) ] by force
qed
end
4 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.

References