Automating Asymptotics in a Theorem Prover

Manuel Eberl

Technical University of Munich

Formal Methods in Mathematics
6 January 2020
My Christmas Project

I found some lovely 5-pages of lecture notes on Transcendental Number Theory by Filaseta:

4 The Irrationality of $\zeta(3)$

For $s > 1$, we define $\zeta(s) = \sum_{n=1}^{\infty} 1/n^s$. We give here a proof by Frits Beukers that $\zeta(3)$ is irrational (the result itself being originally due to R. Apery).

Theorem 10. The number $\zeta(3) = \sum_{n=1}^{\infty} 1/n^3$ is irrational.

In addition to Lemma 1 of the previous section (and the notation given there), we make use of the following results.

Lemma 2. Let $r$ and $s$ be nonnegative integers. If $r > s$, then

$$\int_0^1 \int_0^1 \frac{\log(xy)}{1-xy} x^r y^s \, dx \, dy$$

is a rational number whose denominator when reduced divides $d_r^3$. Also,

$$\int_0^1 \int_0^1 \frac{\log(xy)}{1-xy} x^r y^s \, dx \, dy = 2 \left( \zeta(3) - \sum_{k=1}^{r} \frac{1}{k^3} \right).$$

Proof. Integrating by parts, we obtain that for $k \geq 0$

$$\int_0^1 (\log x)x^{r+k} \, dx = \lim_{\epsilon \to 0} \int_{\epsilon}^{1} (\log x)x^{r+k} \, dx$$
My Christmas Project

So I decided to formalise them:

The Irrationality of $\zeta(3)$

Manuel Eberl

December 28, 2019

Abstract

This article provides a formalisation of Beukers’s straightforward analytic proof [2] that $\zeta(3)$ is irrational. This was first proven by Apéry [1] (which is why this result is also often called ‘Apéry’s Theorem’) using a more algebraic approach. This formalisation follows Filaseta’s presentation of Beukers’s proof [5].

Contents

1 The Irrationality of $\zeta(3)$
   1.1 Auxiliary facts about polynomials .......................... 2
   1.2 Auxiliary facts about integrals ......................... 4
   1.3 Shifted Legendre polynomials ......................... 10
   1.4 Auxiliary facts about the $\zeta$ function .......... 13
   1.5 Divisor of a sum of rationals ......................... 14
   1.6 The first double integral ............................ 16
   1.7 The second double integral .......................... 26
   1.8 The triple integral .................................. 29
   1.9 Connecting the double and triple integral ......... 33
   1.10 The main result ....................................... 44
The Curse of de Bruijn

Mathematical proofs in a proof assistant (compared to pen-and-paper)
The Curse of de Bruijn

Mathematical proofs in a proof assistant (compared to pen-and-paper)

▶ are much longer
The Curse of de Bruijn

Mathematical proofs in a proof assistant (compared to pen-and-paper)

- are much longer
- take more time to write
The Curse of de Bruijn

Mathematical proofs in a proof assistant (compared to pen-and-paper)

▶ are much longer
▶ take more time to write
▶ contain many tedious steps.
The Curse of de Bruijn

Mathematical proofs in a proof assistant (compared to pen-and-paper)

- are much longer
- take more time to write
- contain many tedious steps.

There are many reasons for this.
The Curse of de Bruijn

Mathematical proofs in a proof assistant (compared to pen-and-paper)
  ▶ are much longer
  ▶ take more time to write
  ▶ contain many tedious steps.

There are many reasons for this.

But I want to talk about one in particular.
Externalisation of Work in Paper Proofs

▶ Ambiguities and ‘handwaving’
Externalisation of Work in Paper Proofs

▶ Ambiguities and ‘handwaving’
   In a proof assistant, you have to define everything completely rigorously.
Externalisation of Work in Paper Proofs

- Ambiguities and ‘handwaving’
  In a proof assistant, you have to define everything completely rigorously.

- Side conditions not proven/dismissed as trivial
Externalisation of Work in Paper Proofs

- Ambiguities and ‘handwaving’
  In a proof assistant, you have to define everything completely rigorously.

- Side conditions not proven/dismissed as trivial
  A proof assistant will force you to prove every single side condition.
Externalisation of Work in Paper Proofs

- Ambiguities and ‘handwaving’
  In a proof assistant, you have to define everything completely rigorously.

- Side conditions not proven/dismissed as trivial
  A proof assistant will force you to prove every single side condition.

- A huge trove of ‘library’ results that one can use freely
Externalisation of Work in Paper Proofs

▶ Ambiguities and ‘handwaving’
In a proof assistant, you have to define everything completely rigorously.

▶ Side conditions not proven/dismissed as trivial
A proof assistant will force you to prove every single side condition.

▶ A huge trove of ‘library’ results that one can use freely
Most mathematical results have not been formalised
Externalisation of Work in Paper Proofs

- Ambiguities and ‘handwaving’
  In a proof assistant, you have to define everything completely rigorously.

- Side conditions not proven/dismissed as trivial
  A proof assistant will force you to prove every single side condition.

- A huge trove of ‘library’ results that one can use freely
  Most mathematical results have not been formalised
  And even if: perhaps not in the system you use.
The Curse of de Bruijn

Solution: No idea. :(

Partial solutions: (in my opinion)

▶ Good, concise notation
▶ Good automation

When writing a formal proof, we can externalise work to the reader as well. The reader is the proof assistant.
The Curse of de Bruijn

Solution: No idea. :( 

Partial solutions:
The Curse of de Bruijn

Solution: No idea. :

Partial solutions: (in my opinion)
The Curse of de Bruijn

Solution: No idea. :(

Partial solutions: (in my opinion)
  ▶ Good, concise notation
The Curse of de Bruijn

Solution: No idea. :

Partial solutions: (in my opinion)

- Good, concise notation
- Good automation
The Curse of de Bruijn

Solution: No idea. :(

Partial solutions: (in my opinion)
  ▶ Good, concise notation
  ▶ Good automation

When writing a formal proof, we can externalise work to the reader as well.
The Curse of de Bruijn

Solution: No idea. :(  

Partial solutions: (in my opinion)  
  ▶ Good, concise notation  
  ▶ Good automation  

When writing a formal proof, we can externalise work to the reader as well.  

The reader is the proof assistant.
Domain-Specific Automation

Human mathematicians have a large repertoire of domain-specific automation procedures in their brain:

- How to solve a quadratic equation
Domain-Specific Automation

Human mathematicians have a large repertoire of domain-specific automation procedures in their brain:

- How to solve a quadratic equation
- How to take a derivative
Domain-Specific Automation

Human mathematicians have a large repertoire of domain-specific automation procedures in their brain:

▶ How to solve a quadratic equation
▶ How to take a derivative
▶ How to expand into partial fractions
Domain-Specific Automation

Human mathematicians have a large repertoire of domain-specific automation procedures in their brain:

- How to solve a quadratic equation
- How to take a derivative
- How to expand into partial fractions

This saves lots of time when writing mathematical papers.
Domain-Specific Automation

Human mathematicians have a large repertoire of domain-specific automation procedures in their brain:

- How to solve a quadratic equation
- How to take a derivative
- How to expand into partial fractions

This saves lots of time when writing mathematical papers.

For effective formalisation of mathematics, we need to *teach* proof assistants these skills.
Examples for Domain-Specific Automation

- Cancelling common factors from equations
Examples for Domain-Specific Automation

- Cancelling common factors from equations
- Linear arithmetic (Chaieb/Nipkow)
Examples for Domain-Specific Automation

- Cancelling common factors from equations
- Linear arithmetic (Chaieb/Nipkow)
- Approximation using interval arithmetic (Hölzl)
Examples for Domain-Specific Automation

- Cancelling common factors from equations
- Linear arithmetic (Chaieb/Nipkow)
- Approximation using interval arithmetic (Hölzl)
- Evaluating $\sqrt{16} = 4$ etc.
Examples for Domain-Specific Automation

- Cancelling common factors from equations
- Linear arithmetic (Chaieb/Nipkow)
- Approximation using interval arithmetic (Hölzl)
- Evaluating $\sqrt{16} = 4$ etc.
- Proving primality using Pratt certificates (Wimmer/E.)
Examples for Domain-Specific Automation

- Cancelling common factors from equations
- Linear arithmetic (Chaieb/Nipkow)
- Approximation using interval arithmetic (Hölzl)
- Evaluating $\sqrt{16} = 4$ etc.
- Proving primality using Pratt certificates (Wimmer/E.)
- Evaluating winding numbers (Li)
Examples for Domain-Specific Automation

- Cancelling common factors from equations
- Linear arithmetic (Chaieb/Nipkow)
- Approximation using interval arithmetic (Hölzl)
- Evaluating $\sqrt{16} = 4$ etc.
- Proving primality using Pratt certificates (Wimmer/E.)
- Evaluating winding numbers (Li)
- Real asymptotics (E.)
Automating Real Asymptotics in Isabelle/HOL
Interactive theorem prover; mostly *Higher Order Logic*
Interactive theorem prover; mostly *Higher Order Logic*

Unlike Coq/Lean: *No dependent types*
Interactive theorem prover; mostly *Higher Order Logic*

Unlike Coq/Lean: **No dependent types**

Large library of real and complex analysis
Interactive theorem prover; mostly *Higher Order Logic*

Unlike Coq/Lean: *No dependent types*

Large library of real and complex analysis

Archive of Formal Proofs:
Large collection of Isabelle proof developments
Let’s talk about asymptotics in a proof assistant.
Let’s talk about asymptotics in a proof assistant.

Suppose you write a formal proof and suddenly have to prove

\[ \lim_{x \to \infty} x^2 - x = \infty. \]
Let’s talk about asymptotics in a proof assistant.

Suppose you write a formal proof and suddenly have to prove

\[ \lim_{x \to \infty} x^2 - x = \infty. \]

Any ‘real’ mathematician would rightly dismiss this as trivial.
Let’s talk about asymptotics in a proof assistant.

Suppose you write a formal proof and suddenly have to prove

\[
\lim_{x \to \infty} x^2 - x = \infty.
\]

Any ‘real’ mathematician would rightly dismiss this as trivial.

But in a theorem prover, even something this trivial requires some thinking and several lines of proofs
Let’s talk about asymptotics in a proof assistant.

Suppose you write a formal proof and suddenly have to prove

\[ \lim_{x \to \infty} x^2 - x = \infty. \]

Any ‘real’ mathematician would rightly dismiss this as trivial.

But in a theorem prover, even something this trivial requires some thinking and several lines of proofs.

If you have to do this every 5 minutes, it gets annoying.
Example: Stieltjes constants

\[ \gamma_n = \sum_{k=1}^{\infty} \left( \frac{\ln^n k}{k} - \frac{\ln^{n+1}(k + 1) - \ln^{n+1} k}{n + 1} \right) \]
Example: Stieltjes constants

\[ \gamma_n = \sum_{k=1}^{\infty} \left( \frac{\ln^n k}{k} - \frac{\ln^{n+1}(k + 1) - \ln^{n+1} k}{n + 1} \right) \]

Why does this sum exist?
Example: Stieltjes constants

\[ \gamma_n = \sum_{k=1}^{\infty} \left( \frac{\ln^n k}{k} - \frac{\ln^{n+1}(k+1) - \ln^{n+1} k}{n+1} \right) \]

Why does this sum exist?

Because the summand is \( \sim (k^{-2} \ln^n k) \in O(k^{-3/2}) \)
Example: Stieltjes constants

\[ \gamma_n = \sum_{k=1}^{\infty} \left( \frac{\ln^n k}{k} - \frac{\ln^{n+1}(k + 1) - \ln^{n+1} k}{n + 1} \right) \]

Why does this sum exist?

Because the summand is \( \sim (k^{-2} \ln^n k) \in O(k^{-3/2}) \)
and \( \sum k^x \) is summable for any \( x < -1! \)
Example: Stieltjes constants

\[ \gamma_n = \sum_{k=1}^{\infty} \left( \frac{\ln^n k}{k} - \frac{\ln^{n+1}(k + 1) - \ln^{n+1} k}{n + 1} \right) \]

Why does this sum exist?

Because the summand is \( \sim (k^{-2} \ln^n k) \in O(k^{-3/2}) \) and \( \sum k^x \) is summable for any \( x < -1! \)

But proving those asymptotics by hand is a lot of work.
Example: Lemma required for Akra–Bazzi

\[
\lim_{x \to \infty} \left( 1 - \frac{1}{b \log^{1+\varepsilon} x} \right)^p \left( 1 + \frac{1}{\log^{\varepsilon/2} \left( bx + \frac{x}{\log^{1+\varepsilon} x} \right)} \right) - \\
\left( 1 + \frac{1}{\log^{\varepsilon/2} x} \right) = 0^+
\]
Example: Lemma required for Akra–Bazzi

\[
\lim_{x \to \infty} \left( 1 - \frac{1}{b \log^{1+\varepsilon} x} \right)^p \left( 1 + \frac{1}{\log^{\varepsilon/2} \left( bx + \frac{x}{\log^{1+\varepsilon} x} \right)} \right) - \left( 1 + \frac{1}{\log^{\varepsilon/2} x} \right) = 0^+
\]

Original author: ‘Trivial, just Taylor-expand it!’
lemma akra_bazzi_aux:

filterlim

(\lambda x. (1 - 1/(b*\ln x ^ (1 + \varepsilon)) ^ p) *
(1 + \ln (b*x + x/\ln x ^ (1 + \varepsilon)) ^ (-\varepsilon/2)) -
(1 + \ln x ^ (-\varepsilon/2)))
(at_right 0) at_top
In Isabelle:

```isar
lemma akra_bazzi_aux:
  filterlim
  (λx. (1 − 1/(b * ln x ^(1 + ε))) ^ p) *
    (1 + ln (b * x + x/ln x ^(1 + ε)) ^ (−ε/2)) −
    (1 + ln x ^ (−ε/2)))
  (at_right 0) at_top
```

Omitted: 700 lines of messy proofs
In Isabelle:

\begin{verbatim}
lemma akra_bazzi_aux:
  filterlim
  (λx. (1 − 1/(b * ln x ^(1 + ε)) ^ p) *
      (1 + ln (b * x + x/ln x ^(1 + ε)) ^(-ε/2)) −
       (1 + ln x ^(-ε/2))))
  (at_right 0) at_top
\end{verbatim}

Omitted: 700 lines of messy proofs

Luckily, we now have automation for this:

\begin{verbatim}
by real_asymp
\end{verbatim}
In Isabelle:

```plaintext
lemma akra_bazzi_aux:
  filterlim
  (λx. (1 − 1/(b * ln x ^ (1 + ε))) ^ p) *
  (1 + ln (b * x + x/ln x ^ (1 + ε)) ^ (−ε/2)) −
  (1 + ln x ^ (−ε/2)))
  (at_right 0) at_top
```

Omitted: 700 lines of messy proofs

Luckily, we now have automation for this:

```plaintext
by real_asymp
```

How does it work?
Multiseries Expansions
Multiseries Expansions

Disclaimer: None of this was invented by me.

Related Work:

- Asymptotic Expansions of exp–log Functions
  by Richardson, Salvy, Shackell, van der Hoeven
- On Computing Limits in a Symbolic Manipulation System
  by Gruntz
- Verified Real Asymptotics in Isabelle/HOL
  by E.
Multiseries Expansions

Power series expansions are insufficient for many important functions: $\exp(x)$, $\ln(x)$, $\Gamma(x)$ for $x \to \infty$
Multiseries Expansions
Power series expansions are insufficient for many important functions: \( \exp(x), \ln(x), \Gamma(x) \) for \( x \to \infty \)

Example:
\[
(x + \ln(x))^{-1} \sim \frac{1}{2}x^{-1} - \frac{1}{4}x^{-2} \ln(x) + \frac{1}{8}x^{-3} \ln(x)^2 + \ldots
\]

Solution: Multiseries

- Like an asymptotic power series, but may contain powers of several ‘basis functions’ \( b_1(x), \ldots, b_n(x) \)
Multiseries Expansions

Power series expansions are insufficient for many important functions: \( \exp(x), \ln(x), \Gamma(x) \) for \( x \to \infty \)

Example:
\[
(x + \ln(x))^{-1} \sim \frac{1}{2}x^{-1} - \frac{1}{4}x^{-2} \ln(x) + \frac{1}{8}x^{-3} \ln(x)^2 + \ldots
\]

Solution: Multiseries

- Like an asymptotic power series, but may contain powers of several ‘basis functions’ \( b_1(x), \ldots, b_n(x) \)
- Formally: \( \mathbb{R}[B_1, \ldots, B_n] \) or \( \mathbb{R}[B_n] \ldots [B_1] \)
Multiseries Expansions

Power series expansions are insufficient for many important functions: $\exp(x)$, $\ln(x)$, $\Gamma(x)$ for $x \to \infty$

Example:

$$(x + \ln(x))^{-1} \sim \frac{1}{2}x^{-1} - \frac{1}{4}x^{-2} \ln(x) + \frac{1}{8}x^{-3} \ln(x)^2 + \ldots$$

Solution: Multiseries

- Like an asymptotic power series, but may contain powers of several ‘basis functions’ $b_1(x), \ldots, b_n(x)$
- Formally: $\mathbb{R}[B_1, \ldots, B_n]$ or $\mathbb{R}[B_n] \ldots [B_1]$
- The basis must be ordered descendingly by ‘growth class’: $\forall i. \ln b_{i+1}(x) \in o(\ln b_i(x))$
Multiseries Expansions

Power series expansions are insufficient for many important functions: $\exp(x)$, $\ln(x)$, $\Gamma(x)$ for $x \to \infty$

Example:

$$(x + \ln(x))^{-1} \sim \frac{1}{2}x^{-1} - \frac{1}{4}x^{-2} \ln(x) + \frac{1}{8}x^{-3} \ln(x)^2 + \ldots$$

Solution: Multiseries

- Like an asymptotic power series, but may contain powers of several ‘basis functions’ $b_1(x), \ldots, b_n(x)$
- Formally: $\mathbb{R}[B_1, \ldots, B_n]$ or $\mathbb{R}[B_n] \ldots [B_1]$  
- The basis must be ordered descendingly by ‘growth class’: $\forall i. \ln b_{i+1}(x) \in o(\ln b_i(x))$
- Typical basis: $\exp(x^2), \exp(x), x, \ln x, \ln \ln x$
A coalgebraic view of Multiseries

\texttt{type Basis} = (\mathbb{R} \rightarrow \mathbb{R}) \text{ list}
A coalgebraic view of Multiseries

\textbf{type} Basis = (\mathbb{R} \to \mathbb{R}) \text{ list}

\textbf{datatype} MS : Basis \to Type \text{ where}
  Const : \mathbb{R} \to MS [\ ]
  Series : LList (MS bs \times \mathbb{R}) \to MS (b :: bs)

Additionally: bases and series must be 'sorted'.

Example for a simple operation:

\textbf{negate}:

\textit{negate} (Const c) = Const (-c)

\textit{negate} (Series ts) = Series [(\textbf{negate} c, e) | (c, e) \leftarrow ts]
A coalgebraic view of Multiseries

type Basis  =  (\(\mathbb{R} \rightarrow \mathbb{R}\)) list

datatype MS : Basis \(\rightarrow\) Type where
    Const : \(\mathbb{R} \rightarrow\) MS []
    Series : LList (MS bs \(\times\) \(\mathbb{R}\)) \(\rightarrow\) MS (b :: bs)

Additionally: bases and series must be ‘sorted’.
A coalgebraic view of Multiseries

\textbf{type} Basis \(= (\mathbb{R} \rightarrow \mathbb{R})\) list
\textbf{datatype} MS : Basis \(\rightarrow\) Type \textbf{where}
  Const : \(\mathbb{R} \rightarrow\) MS \([]\)
  Series : LList (MS bs \(\times\) \(\mathbb{R}\)) \(\rightarrow\) MS (b :: bs)

Additionally: bases and series must be ‘sorted’.

Example for a simple operation:
\textbf{negate} : MS bs \(\rightarrow\) MS bs
A coalgebraic view of Multiseries

\textbf{type} Basis = (IR \rightarrow IR) \text{ list}

\textbf{datatype} MS : Basis \rightarrow Type \textbf{ where}
  Const : IR \rightarrow MS []
  Series : LList (MS bs \times IR) \rightarrow MS (b :: bs)

Additionally: bases and series must be ‘sorted’.

Example for a simple operation:

\text{negate} : MS bs \rightarrow MS bs
\text{negate} (\text{Const}\ c) = \text{Const}\ (-c)
A coalgebraic view of Multiseries

```plaintext
type Basis = (IR → IR) list
datatype MS : Basis → Type where
  Const : IR → MS []
  Series : LList (MS bs × IR) → MS (b :: bs)

Additionally: bases and series must be ‘sorted’.

Example for a simple operation:

negate : MS bs → MS bs
negate (Const c) = Const (−c)
negate (Series ts) = Series [(negate c, e) | (c, e) ← ts]
```
More Complicated Operations

- Basic operations (defined corecursively):
  constants, identity, addition, multiplication

- Substitution into convergent power series:
  Gives us division; \( \ln \), \( \exp \), \( \sin \), etc. at non-singular points

- \( \exp \) and \( \ln \) at singular points require specialised procedures and may add new basis elements

- For operations like \( \Gamma \), \( \text{erf} \), \( \text{li} \):
  factor out singularities and treat them separately
More Complicated Operations

- Basic operations (defined corecursively):
  constants, identity, addition, multiplication

- Substitution into convergent power series:
  Gives us division; ln, exp, sin, etc. at non-singular points
More Complicated Operations

- Basic operations (defined co-recursively): constants, identity, addition, multiplication
- Substitution into convergent power series:
  Gives us division; \( \ln \), \( \exp \), \( \sin \), etc. at non-singular points
- \( \exp \) and \( \ln \) at singular points require specialised procedures and may add new basis elements
More Complicated Operations

- Basic operations (defined corecursively):
  constants, identity, addition, multiplication

- Substitution into convergent power series:
  Gives us division; ln, exp, sin, etc. at non-singular points

- exp and ln at singular points require specialised procedures and may add new basis elements

- For operations like Γ, erf, li:
  factor out singularities and treat them separately
Connecting Series and Functions

For simple power series, \( f \sim ts \) can be expressed coinductively:

\[
\begin{align*}
  f(x) \in O(x^e) & \quad f(x) - c \cdot x^e \sim ts \\
  \hline
  f(x) \sim (c, e) \colon ts
\end{align*}
\]
Connecting Series and Functions

For simple power series, \( f \sim ts \) can be expressed coinductively:

\[
\begin{align*}
  f(x) &\in O(x^e) \\
  f(x) - c x^e &\sim ts
\end{align*}
\]

\[
  f(x) \sim (c, e) :: ts
\]

Operations are defined corecursively; correctness is proven coinductively. Both are straightforward.

The same works for multiseries quite similarly.
Finding Expansions

We can construct expansions for functions ‘bottom up’:

1. $\frac{1}{x}$ has the trivial expansion $x - 1$ w.r.t. the basis $[x]$.
2. Substitute the series $x - 1$ into the Taylor expansion of $\sin$.
3. $\exp(x)$ has to be added as a new basis element.
4. $\exp(x)$ then has the trivial expansion $\exp(x)$.
5. Our expansion for $\sin\left(\frac{1}{x}\right)$ must be lifted to the new basis $[\exp(x), x]$.
6. Add expansions for $\sin\left(\frac{1}{x}\right)$ and $\exp(x)$.
Finding Expansions
We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for $\sin\left(\frac{1}{x}\right) + \exp(x)$ for $x \to \infty$:
Finding Expansions
We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for \( \sin\left(\frac{1}{x}\right) + \exp(x) \) for \( x \to \infty \):

- \( \frac{1}{x} \) has the trivial expansion \( x^{-1} \) w. r. t. the basis \([x]\)
Finding Expansions

We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for $\sin\left(\frac{1}{x}\right) + \exp(x)$ for $x \to \infty$:

- $1/x$ has the trivial expansion $x^{-1}$ w. r. t. the basis $[x]$
- substitute the series $x^{-1}$ into the Taylor expansion of $\sin$
Finding Expansions
We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for \( \sin\left(\frac{1}{x}\right) + \exp(x) \) for \( x \to \infty \):

- \( 1/x \) has the trivial expansion \( x^{-1} \) w. r. t. the basis \([x]\)
- substitute the series \( x^{-1} \) into the Taylor expansion of \( \sin \)
- \( \exp(x) \) has to be added as a new basis element
Finding Expansions
We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for \( \sin\left(\frac{1}{x}\right) + \exp(x) \) for \( x \to \infty \):

- \( \frac{1}{x} \) has the trivial expansion \( x^{-1} \) w. r. t. the basis \([x]\)
- substitute the series \( x^{-1} \) into the Taylor expansion of \( \sin \)
- \( \exp(x) \) has to be added as a new basis element
- \( \exp(x) \) then has the trivial expansion \( \exp(x) \)
Finding Expansions

We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for $\sin\left(\frac{1}{x}\right) + \exp(x)$ for $x \to \infty$:

- $1/x$ has the trivial expansion $x^{-1}$ w.r.t. the basis $[x]$
- substitute the series $x^{-1}$ into the Taylor expansion of $\sin$
- $\exp(x)$ has to be added as a new basis element
- $\exp(x)$ then has the trivial expansion $\exp(x)$
- our expansion for $\sin(1/x)$ must be lifted to the new basis $[\exp(x), x]$
Finding Expansions
We can construct expansions for functions ‘bottom up’:

Example

Find an expansion for \(\sin\left(\frac{1}{x}\right) + \exp(x)\) for \(x \to \infty\):

- \(1/x\) has the trivial expansion \(x^{-1}\) w. r. t. the basis \([x]\)
- substitute the series \(x^{-1}\) into the Taylor expansion of \(\sin\)
- \(\exp(x)\) has to be added as a new basis element
- \(\exp(x)\) then has the trivial expansion \(\exp(x)\)
- our expansion for \(\sin(1/x)\) must be \textit{lifted} to the new basis \([\exp(x), x]\)
- add expansions for \(\sin(1/x)\) and \(\exp(x)\)
Finding Expansions

End result: Theorem that $\sin(1/x) + \exp(x)$ has the following expansion w. r. t. basis $(\exp(x), x)$:
Finding Expansions

End result: Theorem that \( \sin(1/x) + \exp(x) \) has the following expansion w.r.t. basis \((\exp(x), x)\):

\[
\text{lift}\_\text{expansion} \left( \text{sin}\_\text{ms} \left( \text{Series} \left[ (1, -1) \right] \right) \right) + \\
\text{Series} \left( \text{Series} \left[ (1, 0) \right], 1 \right)
\]
Finding Expansions

End result: Theorem that \( \sin \left( \frac{1}{x} \right) + \exp(x) \) has the following expansion w.r.t. basis \((\exp(x), x)\):

\[
\text{lift\_expansion} \left( \sin\_ms \left( \text{Series } [(1, -1)] \right) \right) + \text{Series } [(\text{Series } [(1, 0)], 1)]
\]

which evaluates to

\[
\exp(x) + x^{-1} - \frac{1}{6} x^{-3} + \frac{1}{120} x^{-5} - \ldots
\]
Sign Determination

Problem:

- Many operations involve comparisons of real numbers
Sign Determination

Problem:
- Many operations involve comparisons of real numbers
- ‘Trimming’ expansions involves zeroness tests of real functions
Sign Determination

Problem:

- Many operations involve comparisons of real numbers
- ‘Trimming’ expansions involves zeroness tests of real functions
- Both of these are difficult or even undecidable
Sign Determination

Solution: Heuristic approach using Isabelle’s automation
Sign Determination

Solution: Heuristic approach using Isabelle’s automation

- Use automation to determine signs – might fail

Optionally: Use approximation by interval arithmetic

User may have to supply additional facts

This works surprisingly well
Sign Determination

Solution: Heuristic approach using Isabelle’s automation
  ▶ Use automation to determine signs – might fail
  ▶ Use automation to determine if function is identically zero
     – might cause non-termination
Sign Determination

Solution: Heuristic approach using Isabelle’s automation

- Use automation to determine signs – might fail
- Use automation to determine if function is identically zero – might cause non-termination
- Optionally: Use approximation by interval arithmetic
Sign Determination

Solution: Heuristic approach using Isabelle’s automation

- Use automation to determine signs – might fail
- Use automation to determine if function is identically zero – might cause non-termination
- Optionally: Use approximation by interval arithmetic
- User may have to supply additional facts
Sign Determination

Solution: Heuristic approach using Isabelle’s automation

- Use automation to determine signs – might fail
- Use automation to determine if function is identically zero – might cause non-termination
- Optionally: Use approximation by interval arithmetic
- User may have to supply additional facts

This works surprisingly well
Proof Method

With some pre-processing, we can automatically prove statements of the form

- $f(x) \rightarrow c$
- $f(x) \sim g(x)$
- $f(x) < g(x)$ eventually
- $f(x) \in L(g(x))$ for any Landau symbol $L$

as $x \rightarrow l$ for $l \in \mathbb{R} \cup \{\pm \infty\}$
Proof Method

With some pre-processing, we can automatically prove statements of the form

- $f(x) \rightarrow c$
- $f(x) \sim g(x)$
- $f(x) < g(x)$ eventually
- $f(x) \in L(g(x))$ for any Landau symbol $L$

as $x \rightarrow l$ for $l \in \mathbb{R} \cup \{-\infty, \infty\}$

$f$ and $g$ can be built from $+ - \cdot \div \ln \exp \min \max \hat{\cdot} | \cdot | \sqrt[\cdot]{\cdot}$ without restrictions.
Proof Method

With some pre-processing, we can automatically prove statements of the form

\[ f(x) \longrightarrow c \]

\[ f(x) \sim g(x) \]

\[ f(x) < g(x) \text{ eventually} \]

\[ f(x) \in L(g(x)) \text{ for any Landau symbol } L \]

as \( x \to l \) for \( l \in \mathbb{R} \cup \{ \pm \infty \} \)

\( f \) and \( g \) can be built from \( + \ - \ - \ / \ ln \ exp \ min \ max \ \hat{\cdot} \ |\cdot| \ \sqrt{\cdot} \)

without restrictions

\( \text{sin, cos, tan at finite points also possible.} \)
Proof Method

Problem: What about ‘oscillating’ functions like $\sin$, $\lfloor \cdot \rfloor$, mod?
Proof Method

**Problem:** What about ‘oscillating’ functions like \( \sin \), \( \lfloor \cdot \rfloor \), mod?

**Example:**
\[
\sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1)
\]
Proof Method

Problem: What about ‘oscillating’ functions like \( \sin \), \( \lfloor \cdot \rfloor \), \( \text{mod} \)?

Example: \( \sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1) \)

Obvious solution: Asymptotic interval arithmetic:
Proof Method

Problem: What about ‘oscillating’ functions like \( \sin \), \( \lfloor \cdot \rfloor \), mod?

Example: \( \sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1) \)

Obvious solution: Asymptotic interval arithmetic:

\[ \sin x \in [-1; 1] \]
Proof Method

Problem: What about ‘oscillating’ functions like $\sin$, $\lfloor \cdot \rfloor$, mod?

Example: $\sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1)$

Obvious solution: Asymptotic interval arithmetic:

- $\sin x \in [-1; 1]$
- $\lfloor x \rfloor \in [x - 1; x]$
Proof Method

Problem: What about ‘oscillating’ functions like sin, ⌊·⌋, mod?

Example: \( \sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1) \)

Obvious solution: Asymptotic interval arithmetic:

\[
\begin{align*}
\sin x & \in [-1; 1] \\
\lfloor x \rfloor & \in [x - 1; x]
\end{align*}
\]

Result: Pair of asymptotic lower/upper bound with known multiseries expansion
Proof Method

Problem: What about ‘oscillating’ functions like \(\sin, \lfloor \cdot \rfloor, \text{mod}\)?

Example: \(\sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1)\)

Obvious solution: Asymptotic interval arithmetic:

- \(\sin x \in [-1; 1]\)
- \(\lfloor x \rfloor \in [x - 1; x]\)

Result: Pair of asymptotic lower/upper bound with known multiseries expansion

Works in many cases, but does not cope well with cancellations.
Proof Method

Problem: What about ‘oscillating’ functions like \( \sin \), \( \lfloor \cdot \rfloor \), \( \text{mod} \)?

Example: \( \sqrt{\lfloor x \rfloor} = \sqrt{x} + o(1) \)

Obvious solution: Asymptotic interval arithmetic:

\[ \begin{align*}
\sin x & \in [-1; 1] \\
\lfloor x \rfloor & \in [x - 1; x]
\end{align*} \]

Result: Pair of asymptotic lower/upper bound with known multiseries expansion

Works in many cases, but does not cope well with cancellations. Good enough.
Proof Method
Proof Method

Example

\textbf{lemma} \ (\lambda n. \ (1 + 1/n)^n) \longrightarrow \text{exp} \ 1  \\
\textbf{by} \ \text{real\_asymp}
Proof Method

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>lemma ( (\lambda n. (1 + 1/n)^n) \rightarrow \exp 1 ) &lt;br&gt; by real_asymp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>lemma ( (\lambda n. (1 + a/n)^n) \rightarrow \exp a ) &lt;br&gt; by real_asymp</td>
</tr>
</tbody>
</table>
Usage

- ~180 uses of `real_asymp` in the Archive of Formal Proofs
Usage

~180 uses of \texttt{real\_asym} in the \textit{Archive of Formal Proofs}
(most of them by me – but not all of them)
Usage

- ~180 uses of `real_asymp` in the Archive of Formal Proofs (most of them by me – but not all of them)
- Most uses are for fairly trivial examples
Usage

- ~180 uses of real_asymp in the Archive of Formal Proofs (most of them by me – but not all of them)
- Most uses are for fairly trivial examples
- But: Some others would have been quite painful without the method.
Usage

- ~180 uses of `real_asymp` in the Archive of Formal Proofs (most of them by me – but not all of them)
- Most uses are for fairly trivial examples
- **But:** Some others would have been quite painful without the method.
- **And:** The benefit of not having to stop and think about trivialities like \( x^2 - x \to \infty \) should not be underestimated!
My Personal Experience

When formalising some paper and reaching a page full of limits, integrals, and uniform convergence,

I used to feel like this:
My Personal Experience

When formalising some paper and reaching a page full of limits, integrals, and uniform convergence,

I used to feel like this:  
Now I feel like this:
Future Work

What could be improved?
Future Work

What could be improved?

▶ Incomplete support for $\Gamma$, $\psi^n$, erf, arctan
Future Work

What could be improved?

▶ Incomplete support for $\Gamma$, $\psi^{(n)}$, erf, arctan
▶ Zeroness tests could be improved
Future Work

What could be improved?

- Incomplete support for $\Gamma$, $\psi^{(n)}$, erf, arctan
- Zeroness tests could be improved
- Laurent series expansions for complex functions
Future Work

What could be improved?

- Incomplete support for $\Gamma$, $\psi^{(n)}$, erf, arctan
- Zeroness tests could be improved
- Laurent series expansions for complex functions
  $\Rightarrow$ automatic computation of poles, residues, etc.
Questions? Demo?