ThEdu'18

Proving in the Isabelle Proof Assistant that the Set of Real Numbers is not Countable

We present a new succinct proof of the uncountability of the real numbers – optimized for clarity – based on the proof by Benjamin Porter in the Isabelle Analysis theory.

Jørgen Villadsen

18 July 2018

A Verified Simple Prover for First-Order Logic

Jørgen Villadsen, Anders Schlichtkrull & Andreas Halkjær From

PAAR-2018: 6th Workshop on Practical Aspects of Automated Reasoning

17 Pages – 2000 Lines of Isabelle – Soundness and Completeness in 5 Seconds

Code Generation to Simple Rule Language

Here and Now – Isabelle Introductions

1 Slide

2 Slides

3 Slides

Proving in Isabelle that the set of natural numbers N is infinite

Natural numbers 0, 1, 2, …

“Suc n” is “n+1”

Isabelle proof:

Successor function

is not surjective

but is injective

“auto” proof method

```
theory Scratch
  imports Main
begin

theorem
  ‹Suc n ≠ 0›
  and
  ‹n ≠ n' ⟹ Suc n ≠ Suc n'›
  by auto

end
```

Isabelle Primer for Mathematicians

Interactive proof assistants are special programs, which make it possible to check mathematical results up to a nearly absolute level of certainty.

Clearly, computers cannot read and understand natural language, and even if they could, a typical textbook proof usually omits some details and cannot be treated as absolutely rigorous.

To check the proof in an automated proof assistant, you need to write it using a special language, understandable by computers.

This “translation” to computer language is called the formalization of the proof.

In conclusion, the formalization of mathematics in Isabelle is a little bit difficult to start, but very exciting.

After some time, you become comfortable with Isabelle, and then enjoy proving nontrivial theorems to the strongest opponent in the world, who will never overlook your error or non-strict argument.

And maybe, after some time with Isabelle, you also begin to feel, that only formalized theorems are really proved in mathematics.

All the other proofs are just proof outlines.

https://dream.inf.ed.ac.uk/projects/isabelle/

Logic is about formalizing which statements & arguments are valid

$(\lambda x.\ x) = (\lambda y.\ y)$

$A = B \Longrightarrow A \equiv B$

Definitions

$\text{True} \equiv (\lambda x.\ x) = (\lambda x.\ x)$

$\neg P \equiv P \longrightarrow \text{False}$

$\text{False} \equiv (\lambda x.\ x) = (\lambda x.\ \text{True})$

$P \wedge Q \equiv (\lambda x.\ (P \longrightarrow Q \longrightarrow x) \longrightarrow x) = (\lambda x.\ \text{True})$

$P \vee Q \equiv (\lambda x.\ (P \longrightarrow x) \longrightarrow (Q \longrightarrow x) \longrightarrow x) = (\lambda x.\ \text{True})$

Start of the famous incompleteness paper by Kurt Gödel (1931)

The development of mathematics toward greater precision has led, as is well known, to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules...

The Modus Ponens rule in Isabelle

If P ⟶ Q and P then Q

$$(P \longrightarrow Q) \Longrightarrow (P \Longrightarrow Q)$$

```
proposition ‹P ⟶ Q ⟹ P ⟹ Q› by (rule mp)
```

Isabelle Rules

$t = t$

$s = t \Longrightarrow P\ s \Longrightarrow P\ t$

$(\bigwedge x.\ f\ x = g\ x) \Longrightarrow (\lambda x.\ f\ x) = (\lambda x.\ g\ x)$

$(P \Longrightarrow Q) \Longrightarrow P \longrightarrow Q$

$P \longrightarrow Q \Longrightarrow P \Longrightarrow Q$

$P = True \lor P = False$

inj Suc

¬ surj Suc

Formal Proofs of the Uncountability of the Reals

ProofPower	Rob Arthan	2003
Metamath	Norman Megill	2004
Mizar	Grzegorz Bancerek	2004
HOL Light	John Harrison	2005
Isabelle	Benjamin Porter	2005
Coq	Nickolay Shmyrev	2006

```
theory Demo imports Complex_Main begin

theorem ‹∄f. ∀z :: real. ∃n :: nat. f n = z›
proof
  assume ‹∃f. ∀z :: real. ∃n :: nat. f n = z›
  show False
  proof -
    from ‹∃f. ∀z. ∃n. f n = z› obtain f :: ‹nat ⇒ real› where assumption: ‹∀z. ∃n. f n = z› ..

    obtain D :: ‹nat ⇒ real set› where ‹(⋂n. D n) ≠ {}› ‹f n ∉ D n› for n
    proof -
      obtain L R :: ‹real ⇒ real ⇒ real ⇒ real›
        where
          *: ‹L a b c < R a b c› ‹{L a b c .. R a b c} ⊆ {a .. b}› ‹c ∉ {L a b c .. R a b c}›
        if ‹a < b› for a b c
      proof -
        have ‹∃x y. a ≤ x ∧ x < y ∧ y ≤ b ∧ ¬ (x ≤ c ∧ c ≤ y)› if ‹a < b› for a b c :: real
          using that dense less_le_trans not_le not_less_iff_gr_or_eq by (metis (full_types))

        then have ‹∃x y. x < y ∧ {x .. y} ⊆ {a .. b} ∧ c ∉ {x .. y}› if ‹a < b› for a b c :: real
          using that by fastforce

        then show ?thesis
          using that by metis
      qed
```

```
define P :: ‹nat ⇒ real × real›
  where
    ‹P ≡ rec_nat
         (L 0 1 (f 0),
          R 0 1 (f 0))
         (λn (x, y). (L x y (f (Suc n)),
                      R x y (f (Suc n))))›

with *(1) have 0: ‹fst (P n) < snd (P n)› for n
  unfolding split_def by (induct n) simp_all

define I :: ‹nat ⇒ real set›
  where
    ‹I ≡ λn. {fst (P n) .. snd (P n)}›

with 0 have ‹I n ≠ {}› for n
  using less_imp_le by fastforce

moreover from 0 *(2) have ‹decseq I›
  unfolding I_def P_def split_def decseq_Suc_iff by simp

ultimately have ‹finite S ⟶ (⋂n∈S. I n) ≠ {}› for S
  using decseqD subset_empty INF_greatest Max_ge by metis

moreover have ‹closed (I n)› for n
  unfolding I_def by simp

moreover have ‹compact (I n)› for n
  unfolding I_def using compact_Icc compact_Int_closed decseqD inf.absorb_iff2 le0 by simp
```

```
      ultimately have ‹(⋂n. I n) ≠ {}›
        using INT_insert compact_imp_fip_image empty_subsetI finite_insert inf.absorb_iff2 by metis

      moreover from 0 *(3) have ‹f n ∉ I n› for n
        unfolding I_def P_def split_def by (induct n) simp_all

      ultimately show ?thesis ..
    qed

    then obtain e where ‹∄n. f n = e›
      using INT_E UNIV_I ex_in_conv by metis

    moreover from assumption have ‹∃n. f n = e› ..

    ultimately show ?thesis ..
  qed
qed

end
```

We have with good results explained the proof to a group of mathematicians with little or no knowledge of formal methods.

In particular the "…" notation is useful and might be relevant to implement.

We have not yet fully investigated if our approach can be generalized to other proofs except that we have recently considered a related proof, namely that the set of rational numbers is in fact countable, based on the rather scattered formalization in the Isabelle Library which incidentally differs in a number of ways from the traditional proof