Proving Bernoulli's inequality and the Cauchy-Schwarz inequality using Lean

Tianming Zhao, Ximeng Song

23.07.2023

1 Introduction

Inspired by the last level of the Natural Number Game, namely the Inequality World, we decided to prove two of the most important inequalities that we have learned in Analysis I.

Of course, this is only a proof of concept, as these inequalities are obviously already proven. Yet it is still very interesting to formalize the proofs using Lean. We are aware that others have already used Lean to prove these important theorems, as the Lean community is quite vast, and the corresponding library `mathlib` has an extensive collection of already formalized theorems and lemmas - which we will also take advantage of. But we have still decided to challenge ourselves, using our newly acquired skills to formalize the following inequalities.

2 Bernoulli's inequality

The first challenge we want to tackle is Bernoulli's inequality. This is one of the first inequalities that are taught in analysis, as it is very useful to approximate terms of $1 + x$. It can even be used to prove other inequaltiies, such as the inequality of arithmetic and geometric means.

Bernoulli's inequality states the following:

$$\forall x \in \mathbb{R}, \forall n \in \mathbb{N}, -1 \leq x : 1 + nx \leq (1 + x)^n$$

2.1 Proof

We will use induction over n. Let us proof Bernoulli's inequality for the **initial case**, namely $n = 0$.

$$\begin{aligned} 1 + 0 \times x &\leq (1 + x)^0 \\ 1 + 0 &\leq 1 \\ 1 &\leq 1 \end{aligned}$$

As "$\leq$" is a partial order by definition, it is also reflexive, thus concluding the proof for the initial case.

Inductive hypothesis: Let us now assume that $1 + nx \leq (1 + x)^n$ holds true for some n.

Inductive step: Using that both $(1 + x)$ and dx^2 are nonnegative, we have:

$$(1+x)^{n+1} = (1+x)^n(1+x) \geq (1+nx)(1+x) = 1+(n+1)x+nx^2 \geq 1+(n+1)x$$

This concludes the proof by induction.

2.2 Formalizing the proof using Lean

It is obvious that proving Bernoulli's inequality will require not only the theorem statements from the natural numbers but also those from real numbers. Since we don't have the time (and interest) to prove the whole theorem statements of real numbers, we imported the natural number base and real number base from mathlib.

```
import data.nat.basic
import data.real.basic
```

Before we prove the inequality, we introduce a lemma that does not exist in the real number base. We will use this lemma later in the proof of Bernoulli's inequality. The lemma expresses that the multiplication of a natural number with a non-negative real number gives a non-negative real number. Its proof can be easily solved using some basic theorem statements and tactics that we learned from the Natural Number Game.

```
lemma nat_mul_nonneg_real_nonneg (n : \N) (a : \R) :
    a \>= 0 → ↑n * a \>= 0 :=
begin
    intro p,
    have h : n \>= 0 := by {exact nat.zero_le n},
    apply mul_nonneg,
    exact_mod_cast h,
    exact p,
end
```

Now we formalize the inequality in Lean:

```
lemma bernoulli_inequality (x : \R) (p : -1 \<= x)
    (n : \N) : 1 + x * n \<= (1 + x)^n :=
begin
```

Then, we use the `induction` tactic to split the goal into $n = 0$ and $n = succ(d)$. The first part can be easily proven by using the `simp` tactic, which performs all possible `rewrite` tactics to solve the goal.

```
    induction n with d hd,
    simp,
```

Next, we split each step of the original mathematical proof into h1-h4 and rewrite them to the form of our goal. The h1-h4 in mathematical language and the corresponding Lean code to prove them are shown below:

$h1 : (1 + x)^{succ(d)} = (1 + x)^d \times (1 + x)$

```
have h1 : (1 + x) ^ nat.succ d = (1 + x) ^ d * (1 + x) :=
begin
    rw nat.succ_eq_add_one,
    rw pow_add,
    ring,
end,
```

$h2 : (1 + dx)(1 + x) \leq (1 + x)^d(1 + x)$

```
have h2 : (1 + x * d) * (1 + x) \<=
    (1 + x) ^ d * (1 + x) :=
begin
    apply mul_le_mul_of_nonneg_right,
    exact hd,
    have j := add_le_add_right p 1,
    simp at j,
    rw add_comm at j,
    exact j,
end,
```

$h3 : (1 + dx)(1 + x) = 1 + (d + 1)x + dx^2$

```
have h3 : (1 + x * d) * (1 + x) =
    1 + (d + 1) * x + d * x ^ 2 := by ring,
```

$h4 : 1 + (d + 1)x \leq 1 + (d + 1)x + dx^2$

```
have h4 : 1 + ↑(d + 1) * x \<=
    1 + ↑(d + 1) * x + d * x ^ 2 :=
begin
    have j : 0 \<= x ^ 2 := by {exact pow_two_nonneg x},
    have u : 0 \<= ↑d * x ^ 2 := by
    {exact nat_mul_nonneg_real_nonneg d (x ^ 2) j},
    exact le_add_of_nonneg_right u,
end,
```

The upwards arrow ↑ turns the type of the next variable from a natural number into a real number, which avoids the error of type mismatch while multiplying a natural number with a real number. Other theorem statements that we take advantage of are easily understandable by interpreting their names and are similar to the statements of the natural numbers.

To conclude the proof, we will only need to chain h1-h4 using the `rewrite` tactic and the transitive property of "$\leq$":

```
    rw ← h1 at h2,
    rw h3 at h2,
    norm_cast at h2,
    have h5 : 1 + ↑(d + 1) * x \<= (1 + x) ^ nat.succ d :=
        by {exact le_trans h4 h2},
    rw ← nat.succ_eq_add_one at h5,
    rw mul_comm at h5,
    exact h5,
end
```

The Lean Tactic state shows us goals accomplished. We have proven Bernoulli's inequality using Lean.

3 Cauchy Schwarz

After mastering the natural numbers and real numbers, we decide to take a step forward to vector-space and prove the general Cauchy-Schwarz inequality. We choose the version of the inequality that is written using the norm and inner product:

$$\forall \mathbf{v}, \mathbf{w} \in V : |\langle \mathbf{v}, \mathbf{w} \rangle| \leq \|\mathbf{v}\| \|\mathbf{w}\|,$$

where V is an inner product space over some field K.

NOTE: Lean defines the inner product of an inner product space to be semilinear in the first argument instead of the second argument. We will prove the Cauchy-Schwarz inequality using this definition to better follow the proof via Lean.

3.1 Proof

We will first prove the Cauchy-Schwarz inequality for the trivial case $\mathbf{v} = \mathbf{0}$:

$$|\langle \mathbf{v}, \mathbf{w} \rangle| = |\langle \mathbf{0}, \mathbf{w} \rangle| = 0 \leq 0 = \|\mathbf{0}\| \|\mathbf{w}\| = \|\mathbf{v}\| \|\mathbf{w}\|$$

Let now $\mathbf{v} \neq \mathbf{0}$. For these cases we will need a help equation:

$$\|\langle \mathbf{v}, \mathbf{w} \rangle \cdot \mathbf{v} - \langle \mathbf{v}, \mathbf{v} \rangle \cdot \mathbf{w}\|^2 = \|\mathbf{v}\|^2 (\|\mathbf{v}\|^2 \|\mathbf{w}\|^2 - |\langle \mathbf{v}, \mathbf{w} \rangle|^2)$$

This equation can be easily proven using the definition of the norm, the conjurate symmetry of the inner product, as well as the identity $z \times \overline{z} = |z|^2$:

$$\begin{aligned}
\|\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}-\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\|^2 &= \langle\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}-\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w},\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}-\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\rangle\\
&= \langle\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v},\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}\rangle-\langle\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v},\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\rangle\\
&\quad-\langle\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w},\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}\rangle+\langle\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w},\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\rangle\\
&= \langle \mathbf{v},\mathbf{w}\rangle\overline{\langle \mathbf{v},\mathbf{w}\rangle}\langle \mathbf{v},\mathbf{v}\rangle-\langle \mathbf{v},\mathbf{v}\rangle\overline{\langle \mathbf{v},\mathbf{w}\rangle}\langle \mathbf{v},\mathbf{w}\rangle\\
&\quad-\langle \mathbf{v},\mathbf{w}\rangle\overline{\langle \mathbf{v},\mathbf{v}\rangle}\langle \mathbf{w},\mathbf{v}\rangle+\langle \mathbf{v},\mathbf{v}\rangle\overline{\langle \mathbf{v},\mathbf{v}\rangle}\langle \mathbf{w},\mathbf{w}\rangle\\
&= |\langle \mathbf{v},\mathbf{w}\rangle|^2\langle \mathbf{v},\mathbf{v}\rangle-\langle \mathbf{v},\mathbf{v}\rangle|\langle \mathbf{v},\mathbf{w}\rangle|^2\\
&\quad-\langle \mathbf{v},\mathbf{v}\rangle\langle \mathbf{v},\mathbf{w}\rangle\langle \mathbf{w},\mathbf{v}\rangle+\langle \mathbf{v},\mathbf{v}\rangle\langle \mathbf{v},\mathbf{v}\rangle\langle \mathbf{w},\mathbf{w}\rangle\\
&= \langle \mathbf{v},\mathbf{v}\rangle\left(\langle \mathbf{v},\mathbf{v}\rangle\langle \mathbf{w},\mathbf{w}\rangle-\langle \mathbf{v},\mathbf{w}\rangle\overline{\langle \mathbf{v},\mathbf{w}\rangle}\right)\\
&= \|\mathbf{v}\|^2\left(\|\mathbf{v}\|^2\|\mathbf{w}\|^2-|\langle \mathbf{v},\mathbf{w}\rangle|^2\right)
\end{aligned}$$

To prove the Cauchy-Schwarz inequality using this help equation, we first note that the right hand side of the help equation has all the terms of the Cauchy-Schwarz inequality. Also, both $\|\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}-\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\|^2$ and $\|\mathbf{v}\|^2$ are obviously nonnegative. Thus, given that $\|\mathbf{v}\|^2$ is not zero (as we have already proven that case), $\frac{\|\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}-\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\|^2}{\|\mathbf{v}\|^2}$ is also nonnegative, giving

$$0\le\frac{\|\langle \mathbf{v},\mathbf{w}\rangle\cdot\mathbf{v}-\langle \mathbf{v},\mathbf{v}\rangle\cdot\mathbf{w}\|^2}{\|\mathbf{v}\|^2}=\|\mathbf{v}\|^2\|\mathbf{w}\|^2-|\langle \mathbf{v},\mathbf{w}\rangle|^2$$

and further

$$|\langle \mathbf{v},\mathbf{w}\rangle|^2\le\|\mathbf{v}\|^2\|\mathbf{w}\|^2.$$

Since we know that both $|\cdot|$ and $\|\cdot\|$ are nonnegative, taking the square root on both sides gives the Cauchy-Schwarz inequality.

3.2 Formalizing the proof using Lean

First of all, we look up the required theorem statements for the proof in `mathlib` and import the corresponding base, namely the inner product space base.

```
import analysis.inner_product_space.basic
```

Then we define the inner product space in our namespace with all the required characteristics (additivity, commutativity, norm, etc.):

```
namespace cauchy_schwarz

variables {A B : Type*} [is_R_or_C A]
    [normed_add_comm_group B] [module A B]
    [d : inner_product_space.core A B]

local notation ‘norm_sq‘ :=
    @inner_product_space.core.norm_sq A B _ _ _ _

    include d
```

Now let us start with a simple lemma, which states that the squared norm of a vector should be greater than or equal to zero. We will use this lemma later on.

```
lemma zero_le_norm_sq (e : B) : 0 \<= norm_sq e :=
begin
    rw inner_product_space.core.norm_sq,
    exact inner_product_space.core.inner_self_nonneg,
end
```

To prove the inequality, we follow the same procedure as we did in the written proof. Therefore, let us first prove the help equation. This one will use a lot of `rewrite`s, mostly from the inner product space package that we imported at the beginning. In fact, since this equation holds true in general, we will *only* need the `rewrite` tactic to prove it.

```
lemma help_eq (a b : B) : norm_sq (d.inner a b • a -
    d.inner a a • b) = norm_sq a * (norm_sq a * norm_sq b -
    norm (d.inner a b) ^ 2) :=
begin
    rw ← @is_R_or_C.of_real_inj A,
    rw inner_product_space.core.coe_norm_sq_eq_inner_self,
    rw inner_product_space.core.inner_sub_sub_self,
    repeat {rw inner_product_space.core.inner_smul_right},
    repeat {rw inner_product_space.core.inner_smul_left},
    repeat {rw ← mul_assoc},
    rw is_R_or_C.mul_conj,
    rw mul_right_comm,
    rw mul_assoc,
    rw is_R_or_C.mul_conj,
    rw mul_comm,
    rw sub_self,
    rw zero_sub,
    rw add_comm,
    rw ← sub_eq_add_neg,
    rw inner_product_space.core.conj_symm,
    rw mul_right_comm,
    rw mul_comm,
    rw mul_comm (inner a b) (inner a a),
    rw mul_assoc,
    rw ← mul_sub,
    rw ← inner_product_space.core.coe_norm_sq_eq_inner_self a,
    rw ← inner_product_space.core.coe_norm_sq_eq_inner_self b,
    rw mul_comm (inner a b) (inner b a),
    rw ← inner_product_space.core.conj_symm,
    rw is_R_or_C.conj_mul,
    -- rw ← mul_add,
    rw is_R_or_C.norm_sq_eq_def' (inner a b),
    push_cast,
end
```

Now we can finally begin with our proof of the Cauchy-Schwarz inequality:

```
lemma cauchy_schwarz_inequality (x y : B) :
    norm (d.inner x y) \<= norm x * norm y :=
begin
```

Even though we cannot use the good old induction tactic, since the elements of the inequality are vectors, we will still need to prove the inequality for $\mathbf{v} = 0$ first, as shown in the written proof.

```
    by_cases C : x = 0,
    rw C,
    simp,
    exact inner_product_space.core.inner_zero_left y,
```

Now, thanks to the elegant mathematical concept of proof, we can solve the puzzle easily by using a similar strategy as in Bernoulli'S inequality. This involves constantly constructing new hypotheses and transforming them, which correspond to every step in the mathematical proof.

```
    have h1 : 0 \<= norm_sq (d.inner x y • x -
        d.inner x x • y) / norm_sq x :=
    begin
        have j : 0 \<= norm_sq (d.inner x y • x -
            d.inner x x • y) := by {exact zero_le_norm_sq
            (d.inner x y • x - d.inner x x • y)},
        have i : 0 \<= norm_sq x :=
            by {exact zero_le_norm_sq x},
        exact div_nonneg j i,
    end,
    have h2 : norm_sq (d.inner x y • x - d.inner x x • y) /
        norm_sq x = norm_sq x * norm_sq y -
        norm (inner x y) ^ 2 :=
    begin
        rw help_eq,
        rw mul_div_right_comm,
        rw div_self,
        simp,
```

```
        intro r,
        rw inner_product_space.core.norm_sq_eq_zero at r,
        exact C r,
    end,
    rw h2 at h1,
    apply le_of_pow_le_pow 2,
    exact mul_nonneg (norm_nonneg x) (norm_nonneg y),
    simp,
    rw mul_pow,
    have h3 : 0 + norm (d.inner x y) ^ 2 \<= norm_sq x *
        norm_sq y - norm (inner x y) ^ 2 +
        norm (inner x y) ^ 2 := by {apply add_le_add_right h1
        (norm (inner x y) ^ 2)},
    simp at h3,
    repeat {rw inner_product_space.core.norm_sq at h3},
    repeat {rw
        inner_product_space.core.inner_self_eq_norm_mul_norm
        at h3},
    rw pow_two (norm x),
    rw pow_two (norm y),
    push_cast,
    -- exact_mod_cast h3,
    sorry,
end
end cauchy_schwarz
```

You may realize the `sorry` at the end of our proof. This is due to the fact that somewhere along the line, Lean got confused about which norm it should use. This is due to the fact that mathlib has a few different norms that are all the same in an inner product space, but they are not *definitionally* equal to each other. Sadly, even `exact_mod_cast` didn't fix this issue, though one can clearly see that h3 is the same as the Cauchy-Schwarz inequality.

4 Conclusion

Our initial idea was to prove the four famous inequalities in analysis: the Bernoulli inequality, the Cauchy-Schwarz inequality, the arithmetic and geometric inequality, and the Minkowski inequality. However, although the proof of the Bernoulli inequality went smoothly, the Cauchy-Schwarz inequality proved to be very troublesome. Even though it might seem obvious and straightforward, almost every step we tried to realize in Lean ended up with an error at first. This was due to the lack of knowledge of the theorem statements and the vague understanding of the definition of vector-space in Lean. We always looked up the required statements in mathlib (or used `library_search`) right before applying them into the Lean code. Type mismatch, etc. led us to a shortage of time, and therefore we only achieved the first two inequalities of our plan. However, we learned a lot indeed and became way more familiar with Lean than we were after accomplishing the Natural Number Game.