Formalising Mathematics
Project 1

Kexing Ying

January 26, 2022

Introduction

For the first project in *Formalising Mathematics*, I decided to formalise a result from measure theory known as Egorov's theorem. Egorov's theorem is a useful result establishing a relation between convergence almost everywhere and uniform convergence. In particular, Egorov's theorem states that a sequence of almost everywhere convergent functions converge uniformly everywhere except on an arbitrarily small set.

Egorov's theorem is used to prove the Vitali convergence theorem (a generalisation of the monotone convergence theorem for uniformly integrable functions) and is also useful to simplify more elementary results such as convergence almost everywhere implies convergence in measure.

The content of this project has now been published to Lean's maths library - `mathlib` and can be found (at the time of writing) here. As the name of the `mathlib` file suggest, this theorem is intended for future work and is a part of the overall project for formalising martingales in Lean others and I are working towards.

Almost Everywhere Filter

In order to formalise Egorov's theorem, we will need to use the almost everywhere filter from `mathlib` in order to state the statement of convergence everywhere. We will in this section quickly introduce this definition.

`mathlib` defines the almost everywhere filter as the set of all co-null sets (set which complement has measure zero), namely it is defined as (omitting the proofs)

```
def measure.ae {α} [measurable_space α] (μ : measure α) : filter α :=
{ sets := {s | μ sᶜ = 0}, ... }
```

With the above filter defined, one may introduce all the standard definitions involving almost everywhere with this filter. In our use case, we would like to state the condition of convergence almost everywhere. To achieve this, we use the definition

```
def eventually (p : α → Prop) (f : filter α) : Prop := {x | p x} ∈ f
```

Unfolding the definition, we observe that `eventually p μ.ae` is the proposition that $\mu(\{x \mid \mathsf{p}(x)\})^c = 0$, namely, the set of elements not satisfying p has measure 0. For ease of use, we denote the statement `eventually p μ.ae` in Lean as `∀ᵐ x ∂μ, p x`. Thus, with this in mind, we may formulate convergence almost everywhere by taking p the proposition that the sequence of function converges at the point x.

In particular, the proposition that the sequence of functions f_n converges g almost everywhere is represented by the statement

```
∀ᵐ x ∂μ, x ∈ s → tendsto (λ n, f n x) at_top (𝓝 (g x))
```

Formalisation of Egorov's Theorem

We will in this section outline the proof of Egorov's theorem and comment on the formalisation effort. For the remainder of this document, let us assume α is a measure space with measure μ and β is a second-countable metric space.

Theorem 1 (Egorov's Theorem). If $(f_n : \alpha \to \beta)_{n=0}^{\infty}$ is a sequence of measurable functions which converge almost everywhere on a set $s \subseteq \alpha$ of finite measure to $g : \alpha \to \beta$, then, for any $\epsilon > 0$, there exists some $t \subseteq s$ with measure $\mu(t) \leq \epsilon$ and f_n converges uniformly to g on $s \setminus t$.

Proof. Let $\epsilon > 0$ and for $i, j \geq 1$, define

$$C_{ij} := \bigcup_{k=j}^{\infty} \{x \in s \mid |f_k(x) - g(x)| > i^{-1}\}.$$

Since C_{ij} is measurable with finite measure and $C_{i,j+1} \subseteq C_{i,j}$, by the continuity of measures from above, we have

$$\lim_{j \to \infty} \mu(C_{ij}) = \mu\left(\bigcap_{j=1}^{\infty} C_{ij}\right) = 0.$$

Now, since f_k converges to g almost everywhere, by the definition of limits, there exists a subsequence $(C_{i,J(i)})$ such that $\mu(C_{i,J(i)}) < \epsilon 2^{-i}$. Then, by defining

$$t := \bigcup_{i=1}^{\infty} C_{i,J(i)},$$

we have

$$\mu(t) = \mu\left(\bigcup_{i=1}^{\infty} C_{i,J(i)}\right) \leq \sum_{i=1}^{\infty} \mu(C_{i,J(i)}) < \sum_{i=1}^{\infty} \epsilon 2^{-i} = \epsilon.$$

Now, by construction, the elements of t are namely the elements x which $|f_k(x) - g(x)|$ converges slower than i^{-1}, the elements in $s \setminus t$ converges at a speed uniform bounded above by i^{-1}, and hence, f converges to g uniformly on $s \setminus t$ as required. □

In Lean, Egorov's theorem is represented by

```
theorem tendsto_uniformly_on_of_ae_tendsto
  (hf : ∀ n, measurable (f n)) (hg : measurable g)
  (hsm : measurable_set s) (hs : μ s ≠ ∞)
  (hfg : ∀ᵐ x ∂μ, x ∈ s → tendsto (λ n, f n x) at_top (𝓝 (g x)))
  {ε : ℝ} (hε : 0 < ε) :
∃ t ⊆ s, measurable_set t ∧ μ t ≤ ennreal.of_real ε ∧
  tendsto_uniformly_on f g at_top (s \ t)
```

which immediately, by choosing $s =$ `set.univ`, we obtain a version of Egorov's theorem for finite measures (which is arguably more useful especially in probability theory)

```
theorem tendsto_uniformly_on_of_ae_tendsto' [is_finite_measure μ]
  (hf : ∀ n, measurable (f n)) (hg : measurable g)
  (hfg : ∀ᵐ x ∂μ, tendsto (λ n, f n x) at_top (𝓝 (g x)))
  {ε : ℝ} (hε : 0 < ε) :
∃ t, measurable_set t ∧ μ t ≤ ennreal.of_real ε ∧
  tendsto_uniformly_on f g at_top tᶜ
```

To prove this statement several auxiliary definitions were introduced. These are namely,

- `measure_theory.egorov.not_convergent_seq`;
- `measure_theory.egorov.not_convergent_seq_lt_index`;
- `measure_theory.egorov.Union_not_convergent_seq`,

which corresponds to the declarations C_{ij}, $J(i)$ and $\bigcup_{i=1}^{\infty} C_{i,J(i)}$ in the proof above respectively. The implementation of these definitions are mostly identical to their mathematical definitions except perhaps the definition of `not_convergent_seq_lt_index` where we explicitly invoke the axiom of choice.

Recalling that `not_convergent_seq_lt_index` provides the index of the subsequence $J(i)$ of C_{ij} such that $\mu(C_{i,J(i)}) < \epsilon 2^{-i}$ which existence is provided by $\lim_{j\to\infty} \mu(C_{ij}) = 0$, to construct such a function, one needs to use the axiom of choice. More specifically, we will use `classical.some` which depends on the axiom `classical.choice`.

```
def some {a : Sort*} {p : a → Prop} (h : ∃ x, p x) : a := ...
```

In words, given some predicate on some type and the proof that there exists a term of that type satisfying the predicate, `classical.some` chooses an term of that type satisfying the predicate. Hence, armed with `classical.some`, we can construct the required index function by providing a proof that there exists some J_i such that $\mu(C_{i,J_i}) < \epsilon 2^{-i}$. This proof is known as `measure_theory.egorov.exists_not_convergent_seq_lt`.

With these definitions in mind, the proof of the overall statement in Lean is extremely similar to the proof on paper except perhaps for the fact that f converges uniformly on $s \setminus t$. While on paper, we compared the rate of convergence, we will need to fill in the details in Lean. Nonetheless, by unfolding the definitions, this remains to be a routine $\epsilon - \delta$ proof.

Implementation Decisions

There are in general two ways of constructing proofs in analysis such as this one. One may, as this project has demonstrated, define auxiliary definitions and lemmas for the main theorem and prove it modularly. This approach is more readable and is in general easier to work with although it does generate many unuseful declarations. One may make these declarations private although this is normally discouraged since it is possible one might want to reuse them in other contexts, instead, one normally put these declarations in a special namespace to avoid name clashes in the future.

This approach is however not always possible especially if the proof involves inter-winding definitions which carries along a lot of data and proofs. This can be seen from the fact that all lemmas and definitions in this project require a lot of arguments which results in clutter. To avoid this, one may simply define local declarations within the proof itself as all required arguments are then within the local context.

An example of the second approach can be found in `mathlib` in the formalisation of the Lebesgue decomposition theorem here where we see that ξ is defined locally with local lemmas. The downside to this approach is that the proofs becomes excessively long, and normally becomes hard to maintain as the proof will take much longer to compile after each tactic.