Lean in LaTeX with `minted`

Kexing Ying

June 18, 2023

In this short article I will describe how to include Lean snippets in your LaTeX document by using the `minted` package. The simplest method is by including the `minted` package in your document via `\usepackage{minted}` and include Lean code with

```
\begin{minted}{lean}
  <code goes here>
\end{minted}
```

As an example, the following code

```
\begin{minted}{lean}
/-- **The second Borel-Cantelli lemma** -/
lemma measure_limsup_eq_one {s : ℕ → set Ω}
  (hsm : ∀ n, measurable_set (s n)) (hs : Indep_set s μ)
  (hs' : ∑' n, μ (s n) = ∞) :
  μ (limsup s at_top) = 1
\end{minted}
```

will generate the following

```
/-- **The second Borel-Cantelli lemma** -/
lemma measure_limsup_eq_one {s : ℕ → set Ω}
  (hsm : ∀ n, measurable_set (s n)) (hs : Indep_set s μ)
  (hs' : ∑' n, μ (s n) = ∞) :
  μ (limsup s at_top) = 1
```

You might encounter issues when displaying certain unicode resulting in a box with a question mark in your file instead of a unicode. Namely, you might see a □ when your code snippet contains a unsupported unicode. To fix this, you can use the `newunicodechar` package to introduce new unicode characters.

For example, to display the following snippet correctly, we need to include the unicode characters $\mathcal{L}$ and $\mathcal{N}$.

```
\begin{minted}{lean}
lemma unif_integrable_of_tendsto_Lp_zero
  (hp : 1 ≤ p) (hp' : p ≠ ∞) (hf : ∀ n, mem_ℒp (f n) p μ)
  (hf_tendsto : tendsto (λ n, snorm (f n) p μ) at_top (𝓝 0)) :
  unif_integrable f p μ
\end{minted}
```

To achieve this, simply include the following code in your preamble.

```
\usepackage{newunicodechar}
\newunicodechar{𝒩}{\ensuremath{\mathcal{N}}}
\newunicodechar{ℒ}{\ensuremath{\mathcal{L}}}
```

The above code then displays the following

```
lemma unif_integrable_of_tendsto_Lp_zero
  (hp : 1 ≤ p) (hp' : p ≠ ∞) (hf : ∀ n, mem_ℒp (f n) p μ)
  (hf_tendsto : tendsto (λ n, snorm (f n) p μ) at_top (𝓝 0)) :
  unif_integrable f p μ
```

Hence, whenever you encounter a unsupported unicode, simply introduce it with by finding a symbol in math mode which which is similar enough to that unicode and declaring

```
\newunicodechar{<unicode>}{\ensuremath{<math replacement>}}
```

Beyond this, you can also customize the style the code snippet is displayed using options available in `minted`. See this Overleaf article or the official documentation for more details.