# How to express convergence for analysis in Coq

Catherine Lelay

# How to express convergence for analysis in Coq *

Catherine Lelay

Inria Saclay Île-de-France, Toccata, Palaiseau, F-91120
LRI, Université Paris-Sud, CNRS, Orsay, F-91405

Many theorems in analysis are based on the notion of limit. Accordingly, a formalization of convergence in a proof assistant such as Coq must be particularly neat to build a user-friendly library of analysis. Moreover, usual theorems, such as the one below about exchange of limits, use various definitions of limit: limits of sequences (e.g. $\ell_n$) and functions (e.g. $f$), and limits in functional space (uniform convergence of $f_n$ toward $f$).

**Theorem 1** *For all sequences of functions $f_n : A \to F$, with $F$ a complete space, if $f_n$ is uniformly convergent toward $f$ in $A$, $a$ is a limit point of $A$, and $\forall n, f_n \underset{x\to a}{\longrightarrow} \ell_n$, then the sequence $(\ell_n)_n$ converges toward a limit $\ell$ in $F$ and $f \underset{x\to a}{\longrightarrow} \ell$.*

A common approach is to define all the cases independently, but it implies to duplicate some proofs such as $\lim(f+g) = \lim f + \lim g$ which concern both sequences and functions. A better approach is to generalize the notions of limit and neighborhood (Section 1) and the notion of metric space (Section 2). Using Theorem 1 in this general scope, we deduced an unexpected corollary presented in Section 3.

## 1 Generalization of convergence

The first step is the generalization of limits. Indeed, to express a limit $\lim_a f(x) = \ell$, the usual $\varepsilon - \delta$ definition leads to an explosion of cases, depending on whether $a$ is finite or not, and similarly for $\ell$. Some more cases are needed to express limits of sequences too. Our solution is to define convergence using neighborhood:

$$\lim_{x\to a} f(x) = \ell \quad \Leftrightarrow \quad \underbrace{\forall P,\ P \in \mathcal{V}_\ell \Rightarrow f^{-1}(P) \in \mathcal{V}_a}_{\texttt{filterlim}(f,\mathcal{V}_a,\mathcal{V}_\ell)},$$

where $\mathcal{V}_a$ and $\mathcal{V}_\ell$ are neighborhoods of $a$ and $\ell$. We introduce a notation for the neighborhood of an extended real number $a$: `Rbar_locally`$(a)$. The neighborhood of $+\infty$ for natural numbers is named `eventually`. Using this approach, limits of sequences and functions are now two instantiations of a single definition:

$$\lim_{n\to+\infty} u_n = \ell\text{: } \texttt{is\_lim\_seq}(u,\ell) \texttt{ := filterlim}(u, \texttt{eventually}, \texttt{Rbar\_locally}(\ell)) \qquad \text{and}$$
$$\lim_{t\to a} f(t) = \ell\text{: } \texttt{is\_lim}(f,a,\ell) \texttt{ := filterlim}(a, \texttt{Rbar\_locally}(a), \texttt{Rbar\_locally}(\ell)).$$

This definition makes it possible to factorize many proofs, in particular arithmetic operations and composition. Nevertheless, convergence "in the neighborhood of" is not sufficient in Coq because usage of partial functions is unpractical. Inspired by [3], our solution to extend the notion of neighborhood is filters: sets of sets `F : (T → Prop) → Prop` that are stable by intersection and inclusion. These properties abstract all the useful properties of neighborhoods and cover some other notions of convergence such as left and right limits. Filters also make it possible to state Riemann integral as a limit on pointed subdivisions[1]. Indeed, this is the limit of the function $(x, y) \mapsto \sum_i (x_{i+1} - x_i) f(y_i)$ when $\max |x_{i+1} - x_i|$ tends to 0. With the dedicated filter `Riemann_fine` on pointed subdivisions and the neighborhood of a finite value `locally`$(I_f)$, Riemann integral $\int_a^b f(t)\,dt = I_f$ is defined as `is_Rint`$(f, a, b, I_f)$ `:= filterlim`$(f,$ `Riemann_fine`$(a, b),$ `locally`$(I_f))$.

We defined the predicate `Filter` using type classes [2]. This choice allows to automatically instantiate hypotheses of "being a filter" in lemmas. By example, users do not need to provide a proof that two filters given filters `filter_prod F G` is also a filter to use lemmas.

*This research was supported by the Coquelicot project of the Digiteo cluster and the Île-de-France regional council.

[1]pointed subdivisions := pair of finite sequences $\left((x_i)_{0\le i\le n+1}, (y_i)_{0\le i\le n}\right)$ such that $\forall i, x_i \le y_i \le y_{i+1}$

## 2 Generalization of metric spaces

Even if `filterlim` and `Filter` are useful to represent limits in many general spaces, they are not enough to prove Theorem 1. Indeed, the metric of uniform convergence is painful to use: in order to build a finite distance between functions from a distance $d$, we have to use a minimum "$d_\infty(f,g) = \min(\sup_x d(f(x),g(x)),1)$". Yet, in most proofs, the first step is to change "$d_\infty(f,g) < \varepsilon$" into "$\forall x,\ d(f(x),g(x)) < \varepsilon$". The solution to simplify proofs is thus to replace distances $d$ and $d_\infty$ by the predicate `ball : T →R →T →Prop` which is understood as $\mathtt{ball}(x,\varepsilon,y) \Leftrightarrow d(x,y) < \varepsilon$ in simple metric space and $\mathtt{ball}(f,\varepsilon,g) \Leftrightarrow \forall x, d(f(x),g(x)) < \varepsilon$ in functional spaces. The notion of Cauchy convergence and complete space have also been extended to these spaces named `UniformSpace`.

As for filters, we use a Coq mechanism to automatize instantiation. Type classes were not adapted for this structure, then, we use canonical structure [4].

## 3 Exchange of limits and corollaries

Using these new features, the theorem below subsumes Theorem 1 by choosing $F_1 =$ `eventually`

**Theorem 2** *If $V$ is a complete space, $\forall f : T_1 \times T_2 \to V,\ g : T_2 \to V,\ h : T_1 \to V$, $\forall F_1, F_2$ filters,*

$$\lim_{x \to F_1} f(x) = g \wedge \forall x, \lim_{y \to F_2} f(x,y) = h(x) \quad \Rightarrow \quad \lim_{y \to F_2} g(y) = \lim_{x \to F_1} h(x).$$

Thanks to this generalization, we were able to prove the exchange of integral and limit in an unusual way:

$$\lim_{x \to F_1} f(x) = g \wedge \forall x, \int_a^b f(x,y)\,dy = h(x) \quad \Rightarrow \quad \int_a^b g(y)\,dy = \lim_{x \to F_1} h(x).$$

Indeed, Riemann integral is a limit, we can use Theorem 2 with $F_2 =$ `Riemann_fine`$(a,b)$ and difficulties of this proof are reduced to proving that uniform convergence of the function $f$ toward $g$ imply the convergence of Riemann sums. It also proves that continuity of $g$ implies integrability by choosing $f$ a family of step functions that uniformly converge toward $g$.

As the filter $F_1$ is already general, it can be applied with $F_1 =$ `eventually` to obtain Theorem 1. It also can be used to prove that continuity implies integrability in a complete space with a specific filter on positive real.

## 4 Conclusion

This generalization of limits and metric spaces comes with a comprehensive development of analysis available at `http://coquelicot.saclay.inria.fr/`, compatible with the standard library `Reals`. Most of the theorems can be applied as easily for sequences, real functions, complex functions, vector functions, and so on. More details are available in [1].

## References


[1] Sylvie Boldo, Catherine Lelay, and Guillaume Melquiond. Coquelicot: A User-Friendly Library of Real Analysis for Coq. 2013. Accepted in Mathematics in Computer Science. `http://hal.inria.fr/hal-00860648`.

[2] Pierre Castéran and Matthieu Sozeau. A gentle introduction to type classes and relations in Coq. Technical Report hal-00702455, version 1, 2012.

[3] Johannes Hölzl, Fabian Immler, and Brian Huffman. Type Classes and Filters for Mathematical Analysis in Isabelle/HOL. In *Interactive Theorem Proving*, volume 7998 of *Lecture Notes in Computer Science*, pages 279–294. 2013.

[4] Assia Mahboubi and Enrico Tassi. Canonical Structures for the working Coq user. In Sandrine Blazy, Christine Paulin, and David Pichardie, editors, *Proceeding of the 4th International Conference of Interactive Theorem Proving (ITP 2013)*, volume 7998 of *Lecture Notes in Computer Science*, pages 19–34, Rennes, France, July 2013.