Formalization of Ostrowski theorems in Lean theorem prover

Ryan Lahfa[†,1,*] Julien Marquet[†,1] Hadrien Barral[1]

Abstract

Ostrowski theorems provide classification of all absolute values in certain fields and lies at the foundations of Berkovich space theory. In particular, over $\mathbb{Q}$, all absolute values are either the trivial, the usual or a p-adic. This statement entirely determines the Berkovich spectrum of integers.

We formalize Ostrowski theorems in the Lean theorem prover, in two attempts, one aiming to understand the challenges and determining a reachable generalization target. The second attempt reaches this target and shows everything the first attempt does in a simpler and cleaner way. Following this road, we identify low-hanging fruits missing the Lean mathematical library and develop a self-contained reusable general theory to formalize Ostrowski theorems in general contexts. Our proofs show the discrepancy between how easy it is to use algebra versus how tedious it is to conduct analytical reasoning with inequalities and calculus, and calls for a thorough examination on how to drastically simplify analysis in these contexts.

[†] These authors contributed equally to this work.

[1] DIENS, École Normale Supérieure, CNRS, PSL University, Paris, France

[*] Correspondence: Ryan Lahfa <ryan.lahfa@ens.fr>

1 Introduction

1.1 Background work

The formalization of mathematics has seen a lot of projects: [Wie94], [Ber+21], [Gon+13], [BCM20], [Lew19], [CL21], most of them treat of undergraduate mathematics and seldom of research-level mathematics. In particular, the surrounding mathlib [Com20] formalization projects are progressing at a fast pace, with Witt vectors [CL21], schemes [Buz+21], the Liquid Tensor Experiment [Sch21]. Yet, formalizing research-level theories remain very difficult, especially when the theory requires non-trivial metaprogrammation and tactics to simplify proof terms.

In [BCM20], the definition of perfectoid spaces is formalized entirely and required 33 files and more than 3000 lines of code which should have been in the mathematical library (so-called `for_mathlib` folder), upstreaming back such amount of contributions is also a non-trivial problem [DEL20]. Their formalization also used ad-hoc automation, notably with non-classical objects like algebraic structure "with zero".

In this paper, we will formalize the very start of an alternative theory: Berkovich spaces. This paper follows those ideas and provides an attempt to open up a formalization of Berkovich's young theory. To the best of our knowledge, this formalization has never made its way in any proof assistant.

We also show along the way that picking up a research-level theory produces many undergraduate-level theorems the Lean mathematical library lacks and how it can provide for better interfaces for further formalizations.

In: A. Editor: Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

1.2 Ostrowski theorem and Berkovich spaces

This work will provide an in-depth view on the process of formalizing Ostrowski theorem and its variants. In this section, we will first re-introduce the mathematical contents. In section 2, we detail our bruteforce attempt to formalizing the basic version of the theorem with minimal tooling. In section 3, we use lessons learnt from the previous section to generalize our tooling so that the Ostrowski theorem and its variants can be derived while reusing as much as possible the steps and arguments. In section 4, we provide our feedback on the process and discuss future work to improve such formalizations and this work.

The core objects of Ostrowski's theorem are **absolute values**:

Definition 1 (absolute value)**.** *An absolute value on a ring R is a function $|\cdot| : R \to \mathbb{R}$ such that*

1. $\forall x \in R, |x| = 0 \iff x = 0$
2. $\forall x, y \in R, |xy| = |x||y|$
3. $\forall x, y \in R, |x + y| \leq |x| + |y|$

The usual absolute value is an absolute value with respect to Definition 1.

These objects allow to build a completion of $\mathbb{Q}$ in an algebraically interesting way. The usual completion of $\mathbb{Q}$ is $\mathbb{R}$, and is obtained with the usual absolute value. Absolute values retain just the right amount of properties of the usual absolute value to show *both* analytical and algebraic interest.

In this paper, we focus on the following class of absolute values:

Definition 2 (p-adic absolute value)**.** *With $p \in \mathbb{N}$ prime, we denote $|\cdot|_p$ the p-adic absolute value on $\mathbb{Z}$, where $v_p(k)$ is the multiplicity of p in k:*

$$|k|_p = p^{-v_p(k)}$$

The superclass of p-adic absolute values is the class of the *non-Archimedean absolute value*:

Definition 3 (non-Archimedean absolute value)**.** *An absolute value $|\cdot|$ is called* non-Archimedean *when the following holds:*

$$\forall x, y \in R, |x + y| \leq \max(|x|, |y|)$$

A natural question is to classify all absolutes values over $\mathbb{Q}$, which are classified *up to equivalence*:

Definition 4 (equivalence)**.** *Two absolute values $|\cdot|_1$ and $|\cdot|_2$ on a ring R are said to be* equivalent *when for some $\alpha > 0$ we have $\forall x \in R, {|x|_1}^{\alpha} = |x|_2$.*

When this holds, we write $|\cdot|_1 \sim |\cdot|_2$.

It is noteworthy that equivalent absolute values are topologically equivalent: this turns Ostrowski's theorem into a bridge between algebra and analysis, completely classifying the absolute values on $\mathbb{Q}$.

Theorem 1 (Ostrowski)**.** *Given $\lambda : \mathbb{Q} \to \mathbb{Q}$ an absolute value over $\mathbb{Q}$, either $\lambda \sim |\cdot|$, either there is some $p \in \mathbb{P}$ such that $\lambda \sim |\cdot|_p$.*

Such a theorem shows there is an alternative to the completion of $\mathbb{Q}$ by taking a prime number p and completing using p-adic absolute value, giving arise to $\mathbb{Q}_p$, and that these completions are the only alternatives to the usual one.

Ostrowski's theorem plays an interesting role in Berkovich space theory to completely determine the structure of the Berkovich spectrum of integers: $\mathcal{M}(\mathbb{Z})$, which is the set of all norms over $\mathbb{Z}$ equiped with a certain topology.

Note that Ostrowski theorem has many variants where we can extend it to fields like $\mathbb{F}[X]$ or more complex structures. We will explore in this work how a formalization of multiple variants can be obtained efficiently.

For a more in-depth presentation of Berkovich space theory, refer to [DFN15] or [Tem15].

2 Naive formalization

To understand the challenges behind Ostrowski theorem being formalized, we attempted a bruteforce formalization over $\mathbb{Q}$ based on [Rui19].

The resulting proof is easily understandable, only basic mathematical tooling was needed as in the original proof: Bézout's identity, simple limits and calculus.

Yet this proof does not fit the standard of formalized mathematics: it is far too long and would greatly benefit from:

- extraction of lemmas, and generalization of most parts,
- automation: most of the proof is calculus and could be automated with the right tactics and systems.

Concretely, the core lemma of this first attempt is around 200 lines long. It is built mainly with the `obtain` keyword, which is the formal equivalent of saying "let us now show that ...". This construct allows us to stay close to the intuition but led us to longer proofs, like in the toy example that follows.

For instance, one would start the proof of the bounded case by "let us first show that there is some $n \in \mathbb{N}, n > 0$ such that $|n| < 1$" (in this context, $|\cdot|$ is a nontrivial bounded absolute value). To quickly prove this statement on a piece of paper, we may say that:

- assuming $\forall n \in \mathbb{N}^*, |n| \geq 1$, then $\forall n \in \mathbb{N}^*, |n| = 1$ ($|\cdot|$ is bounded),
- this is absurd because by hypothesis, $|\cdot|$ is nontrivial.

Following this exact scheme, our formalized proof starts with the following :

```
obtain ⟨ n, zero_lt_n, abvn_lt_one ⟩: ∃ n: ℕ, 0 < n ∧ abv n < 1,
{ /- 18 lines omitted -/ }
```

Suddenly, a two line "human" proof came out as a 18 lines long formalized version. In fact, what we really did when we proved this property in two sentences was:

- proceed by *reducio ad absurdum*,
- realize that $|\cdot|$ is equal to 1 everywhere on $\mathbb{N}$,
- prove it by bounding the values of $|\cdot|$ using the suitable hypotheses,
- realize that this is actually enough to prove that $|\cdot|$ is trivial,
- show the contradiction by recalling our hypothesis : $|\cdot|$ is nontrivial.

Formalizing our two-liner required getting into punctilious details, and even further formal considerations when detailing the very informal "realize that ...". Our readers can easily imagine how a handful of calculations became a 200 lines formal proof for the core lemma.

3 Pursuing a general enough point of view

Naturally, the previous proof lacked of generality and contained too much irrelevant detail which translated into bothersome ad-hoc statements, so we adopted two objectives from this experience:

- as much as possible, make Ostrowski theorem a natural consequence from the general theory and allow for interesting generalizations, e.g. Ostrowski over $\mathbb{F}[X]$,
- see how to fit parts of this general theory in the Lean mathematical library, so it can benefit other users.

Our intuition is a synthetic point of view is more suitable for formalization than an analytic approach. Therefore, we went looking for the adequate algebraic theories to support our goals.

We take inspiration from [Art05] presentation of Ostrowski theorem and transform the approach in a suitable way for formalization.

3.1 Core of the theory

For this presentation, we will use R a principal ideal domain (PID).

The core idea is to keep an algebraic point of view and develop some tools to characterize the behavior of bounded absolute values on general rings (Definition 5).

Definition 5. *Given $|\cdot| : R \to \mathbb{R}$ an absolute value, $|\cdot|$ is said bounded when:*

$$\forall x \in R, |x| \leq 1$$

Note that this is equivalent to the usual definition of boundedness (existence of some upper bound):

- if $|\cdot|$ is bounded, then 1 is an upper bound,
- otherwise there is some x such that $|x| > 1$, then $|x^n| \xrightarrow[n \to +\infty]{} +\infty$ and $|\cdot|$ has no finite upper bound.

Furthermore, we define the *trivial absolute value* as the function that maps 0 to 0 and any other element to 1.

We will need one extra lemma for the core theorem, stating that an absolute value is bounded over $\mathbb{N}$ if and only if it is non-Archimedean:

```
theorem nonArchimedean_iff_integers_bounded
  {α} [comm_ring α] [nontrivial α] (abv: α → ℝ) [is_absolute_value abv]:
  (∃ C: ℝ, 0 < C ∧ ∀ n: ℕ, abv n ≤ C) ↔ (∀ a b: α, abv (a + b) ≤ max (abv a) (abv b))
```

Proving this lemma revealed to be challenging: on the paper, it takes at most a dozen of lines, but the formalization took around 200 lines. The reasons are the same as in section 2. We have isolated a corner of the theory where calculus cannot be avoided, like we moved the problem that lied in section 2 from one place to another. As future work, these lines would greatly benefit from new calculus tactics.

The main theorem is `abv_bounded_padic`, which states that a non-trivial bounded absolute value on a principal ideal domain R is a p-adic absolute value for some prime p of R.

```
theorem abv_bounded_padic {α} [integral_domain α] [is_principal_ideal_ring α]
    [normalization_monoid α]
    (abv: α → ℝ) [is_absolute_value abv]
    (bounded: ∀ a: α, abv a ≤ 1)
    (nontrivial: ∃ a: α, a ≠ 0 ∧ abv a ≠ 1):
      ∃ (p: α) (p_prime: prime p), abvs_equiv abv (sample_padic_abv p p_prime)
```

The typeclasses `[integral_domain α]` and `[is_principal_ideal_ring α]` ensure that α is a principal integral domain (PID).

`[normalization_monoid α]` means that the elements of α admit a normal form (say, in $\mathbb{Z}$, the positive integers, and in $\mathbb{K}[X]$, the monics). This is required by some of the lemmas we use, but can be omitted for the scope of this paper.

`abvs_equiv` is the relation of equivalence between absolute values.

`sample_padic_abv p p_prime` is an p-adic absolute value (`p_prime` is a proof that p is indeed prime).

Keeping in mind that according to `nonArchimedian_iff_integers_bounded`, $|\cdot|$ is non-Archimedian, the strategy to prove the core lemma (`abv_bounded_padic`) is as follows:

- Take $\{x \in R \mid |x| < 1\}$, this is a prime ideal of R;
- As R is a PID, there is some prime $p \in R$ that generates the previous set;
- Now, it is sufficient to prove the equivalence between $|\cdot|$ and $|\cdot|_p$ to finish;
- By the primes extensionality lemma (see 3.2), it suffices to prove there is some $\alpha > 0$ such that for all prime $q \in R$, $|q|^\alpha = |q|_p$;
- To clear this goal, a case analysis on whether p and q are associated is enough and helps to find the suitable α in terms of logarithms of absolute values of p.

The core lemma is easy to prove as it is the result of composable and reusable lemmas and proves our point regarding the need of finding general enough abstractions so that the proofs tend towards an assembling game.

3.2 A lemma

We also encountered a very useful extensionality lemma for morphisms over monoids with zero, of which we give the Lean definition:

```
theorem ext_hom_primes {α} [comm_monoid_with_zero α] [wf_dvd_monoid α]
  {β} [monoid_with_zero β]
  (φ₁ φ₂: monoid_with_zero_hom α β)
  (h_units: ∀ u: units α, φ₁ u = φ₂ u)
  (h_irreducibles: ∀ a: α, irreducible a → φ₁ a = φ₂ a):
    φ₁ = φ₂
```

`[monoid_with_zero β]` states that β is a monoid that contains a "zero", *i.e.* an absorbing element. These objects may seem peculiar to a mathematician, but are useful in the context of formalized mathematics. We will not discuss the use of "monoids with zero", as they are outside of the scope of this article.

`[comm_monoid_with_zero α]` further states that α is commutative.

φ `: monoid_with_zero_hom α β` states that φ is a homomorphism of monoid with zero with source α and target β.

`[wf_dvd_monoid α]` states that the division on α is a well-founded order. This is key to the lemma: we only need to proceed by induction. This makes this lemma apply well to principal ideal domains, because the division in such rings the inclusion is well-founded.

Mathematically, this lemma states that if

- R is a principal ideal domain
- Two multiplicative functions agree on the units of R and on its primes

Then, they coincide everywhere.

This nontrivial lemma may be useful to anyone working with multiplicative functions and was added to mathlib [Com20]. We therefore fulfilled one of our two goals: formalizing mathematics which may be useful to future users.

We brought the problem into statements about multiplicative functions, but have yet to lift our original statement which has a valuation flavor in these terms. Note that valuations and multiplicative functions (actually, homomorphisms) are unfortunately very different objects in Lean: the former are just functions that are refined using a typeclass, while the latter are *structures* (in a nutshell, tuples containing objects and proofs). This implies that switching from the valuation point of view to homomorphisms and back is cumbersome. To solve this problem, we had to write some boilerplate to bridge this gap. As future work, it might be possible to automate the process of switching of point of view on this kind of objects, certainly through meta-programming [CL21].

3.3 Application: Ostrowski on $\mathbb{Q}$

Once the core lemmas are laid out, Ostrowski's theorem on $\mathbb{Z}$ is almost immediate. Now, obtaining it over $\mathbb{Q}$ requires the extension of absolute values to the entire field. In theory, it is also almost immediate because of the multiplicative property of absolute values.

In practice, some manual work remained to lift results from $\mathbb{Z}$ to $\mathbb{Q}$, yet, this is not a failure of the previous goal to pursue a general enough theory but rather what we would believe to be a lack of automation in the proof assistant which could be alleviated by meta-programming. That being said, we did not pursue this venture and test our hypothesis and will discuss it later.

3.4 Application: Ostrowski on $\mathbb{F}[X]$

We proved a statement that is slightly less powerful in spirit, in that it does not actually cover *all* the possible absolute values, but only the absolute values that are trivial on $\mathbb{F}$.

Theorem 2 (Ostrowski variant)**.** *Given $|\cdot|$ an absolute value on $\mathbb{F}[X]$, trivial on $\mathbb{F}$. Exactly one of the following is true:*

- *$|\cdot|$ is bounded and for some prime $p \in \mathbb{F}[X]$, $|\cdot| = |\cdot|_p$.*
- *$|\cdot|$ is equivalent to the degree.*

Comforting our intuition, both cases were straightforward reusing the tools in section 3.1.

4 Conclusion

4.1 Results

We wanted to examine the difficulties of formalizing Ostrowski's theorem which constitutes the first step towards Berkovich space theory formalization.

With a bruteforce method, we encountered many tedious computations of analytical nature which led us to hide all the complexity inside algebra which was easier to handle in the proof assistant. The second part presents an approach which worked effectively and gave us with fewer efforts more theorems and provided us with insights on how to pursue the generalization.

Nevertheless, this suggests that calculus and analysis might benefit from a local framework which might help with their manipulation, non-standard analysis seems a promising avenue already explored in Isabelle/HOL with [Fle00], but also, the Lean theorem prover in its version 4 might help with its treatment of coercions and performance improvements [MU21].

4.2 Outlook

We notice formalization is not only a process that helps you verify a proof but also to understand and provide insights on results surrounding a theory and sustain the improvements on the system being used beyond classical computer science aspects like performance or user experience.

In particular, we identified pain points using the Lean theorem prover which constitutes interesting future works, namely automation to:

- combine analysis and inequalities/equalities reasoning, e.g. taking limits on a side or both sides,
- bridge points of view or even theories, *e.g.* the former discussion on valuations and homomorphisms.

Despite these bothersome points, we found that adopting a synthetic approach alleviates us from most of the hardships that we encountered with an analytic approach.

Finally, now that Ostrowski theorems are formalized, it is possible to produce the basic objects of Berkovich space theory, notably the Berkovich spectrum and give a non-trivial example: $\mathcal{M}(\mathbb{Z})$.

References

[Art05] Emil Artin. *Algebraic Numbers and Algebraic Functions*. 2005. ISBN: 978-0-8218-4075-7.

[BCM20] Kevin Buzzard, Johan Commelin, and Patrick Massot. "Formalising Perfectoid Spaces". In: *Proceedings of CPP 2020*. 2020. DOI: 10.1145/3372885.3373830.

[Ber+21] Sophie Bernard et al. "Unsolvability of the Quintic Formalized in Dependent Type Theory". In: (2021).

[Buz+21] Kevin Buzzard et al. *Schemes in Lean*. 2021. arXiv: 2101.02602 [math.AG].

[CL21] Johan Commelin and Robert Y. Lewis. "Formalizing the Ring of Witt Vectors". In: *Proceedings of CPP 2021*. 2021. DOI: 10.1145/3437992.3439919.

[Com20] The mathlib Community. "The lean mathematical library". In: *Proceedings of CPP 2020* (2020). DOI: 10.1145/3372885.3373824.

[DEL20] Floris van Doorn, Gabriel Ebner, and Robert Y. Lewis. "Maintaining a Library of Formal Mathematics". In: *Intelligent Computer Mathematics* (2020). DOI: 10.1007/978-3-030-53518-6_16.

[DFN15] Antoine Ducros, Charles Favre, and Johannes Nicaise, eds. *Berkovich Spaces and Applications*. Vol. 2119. Lecture Notes in Mathematics. 2015. DOI: 10.1007/978-3-319-11029-5.

[Fle00] Jacques D Fleuriot. "On the mechanization of real analysis in Isabelle/HOL". In: *International Conference on Theorem Proving in Higher Order Logics*. 2000.

[Gon+13] Georges Gonthier et al. "A Machine-Checked Proof of the Odd Order Theorem". In: *Interactive Theorem Proving*. Lecture Notes in Computer Science. 2013. DOI: 10.1007/978-3-642-39634-2_14.

[Lew19] Robert Y. Lewis. "A Formal Proof of Hensel's Lemma over the p-Adic Integers". In: *Proceedings of CPP 2019*. 2019. DOI: 10.1145/3293880.3294089.

[MU21] Leonardo de Moura and Sebastian Ullrich. "The Lean 4 Theorem Prover and Programming Language". In: *Automated Deduction – CADE 28*. 2021. ISBN: 978-3-030-79876-5.

[Rui19] Joshua Ruiter. "Ostrowski's Theorem and Completions of Fields". In: (2019).

[Sch21] Peter Scholze. "Liquid tensor experiment". In: *Experimental Mathematics* (2021).

[Tem15] Michael Temkin. "Introduction to Berkovich Analytic Spaces". In: *Berkovich Spaces and Applications*. 2015. DOI: 10.1007/978-3-319-11029-5_1.

[Wie94] Freek Wiedijk. "The QED Manifesto Revisited". In: (1994), p. 14.