Formalizing the Ring of Witt Vectors

Johan Commelin
jmc@math.uni-freiburg.de
Albert–Ludwigs-Universität Freiburg
Freiburg, Germany

Robert Y. Lewis
r.y.lewis@vu.nl
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands

Abstract

The ring of Witt vectors $\mathbb{W}R$ over a base ring R is an important tool in algebraic number theory and lies at the foundations of modern p-adic Hodge theory. $\mathbb{W}R$ has the interesting property that it constructs a ring of characteristic 0 out of a ring of characteristic $p > 1$, and it can be used more specifically to construct from a finite field containing $\mathbb{Z}/p\mathbb{Z}$ the corresponding unramified field extension of the p-adic numbers $\mathbb{Q}_p$ (which is unique up to isomorphism).

We formalize the notion of a Witt vector in the Lean proof assistant, along with the corresponding ring operations and other algebraic structure. We prove in Lean that, for prime p, the ring of Witt vectors over $\mathbb{Z}/p\mathbb{Z}$ is isomorphic to the ring of p-adic integers $\mathbb{Z}_p$. In the process we develop idioms to cleanly handle calculations of identities between operations on the ring of Witt vectors. These calculations are intractible with a naive approach, and require a proof technique that is usually skimmed over in the informal literature.

Keywords: formal math, ring theory, number theory, Lean

1 Introduction

ryl: This is a draft version!

To do: fix overfull and underfull lines once all changes are made

Formalizing a full undergraduate mathematics curriculum has long been a goal of the proof assistant community [31]. This horizon is arguably now in sight: most topics in the standard curriculum can be found in at least one major proof assistant library. As researchers, though, we cannot simply take this as a win. With undergraduate mathematics done we must turn to new challenges.

Formalizations of modern research mathematics are laudable, but remain rare for good reason. Such projects tend to take massive efforts [10, 13], to formalize only part of the main result [28], or to target theorems that are exceptionally well-suited for mechanization [7].

This scarcity of results is hardly surprising. Mastery of undergraduate topics is necessary to do research mathematics, but far from sufficient: Buzzard, Commelin, and Massot [3] note the depth of theory that is needed even to define the structures studied in many subfields. We may be nearing the first horizon of undergraduate mathematics, but the sea between us and the second horizon—graduate mathematics—is vast, little explored, and filled with adventures.

As a new expedition into this sea, we have constructed the ring of p-adic Witt vectors and related operations in the Lean proof assistant and verified some of their fundamental properties. Specifically, we define the Teichmüller lift and the Frobenius and Verschiebung operators, and show that the ring of Witt vectors over $\mathbb{Z}/p\mathbb{Z}$, the integers modulo p, is isomorphic to the p-adic integers. To our knowledge, these topics have never before been formalized in a proof assistant. Our development pushes forward the front line of formalizations in ring theory.

Our project resulted in substantial additions to the ring theory and multivariate polynomial sections of Lean's mathematical library mathlib [21]. Building on Lewis' development of the analytic properties of the p-adic numbers $\mathbb{Q}_p$ and p-adic integers $\mathbb{Z}_p$ [17], we have established more of their algebraic properties: we show that $\mathbb{Z}_p$ is a discrete valuation ring and is the projective limit of the rings $\mathbb{Z}/p^n\mathbb{Z}$ of integers modulo p^n. Our project also served to stress test Lean 3's type class inference mechanism in an algebraic context.

The early theory of Witt vectors was developed in the 1930s [25, 32]. They form a fundamental tool in algebraic number theory and lie at the foundations of modern p-adic Hodge theory. For example, they provide an elegant way to construct unramified $\mathbb{Z}_p$-algebras with prescribed finite residue fields of characteristic p. The ring of Witt vectors also appears in the definitions of Fontaine's period rings [9] that are important in the classification of p-adic Galois representations. Indeed, all the ingredients for the definition of the period ring B_{dR} have now been formalized in Lean.

Witt vectors have a reputation among mathematicians of being rather forbidding and impenetrable. Presentations often skip the details of technical proofs and lengthy calculations. One would reasonably expect a formalization to be even more forbidding, but we have tried in our development to find idioms that lead to short, clean proofs and calculations. This has generally been a success: in many cases, we have been able to reduce goals to universal calculations in the language of rings (Section 4.4), which can be discharged by very simple tactics (Section 5.2). We believe that these statements and proofs are mathematically legible. We were not able to erase the details entirely, though. Our proofs that certain polynomials are integral are long, slow, and unreadable.

Woodstock '18, June 03–05, 2018, Woodstock, NY
2018. ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Some components of our project have already been integrated into mathlib and we expect to integrate the rest over time. For the sake of anonymous review, we provide as an artifact a copy of mathlib with a map to our main additions, where we have removed our names from new files.

2 Preliminaries

The contributions described in this paper can be roughly split into three parts:

1. We expand the algebraic theory of the ring of p-adic integers $\mathbb{Z}_p$.
2. We define the notion of a Witt vector over an arbitrary ring R and construct a ring structure on the set of Witt vectors itself, additionally defining some fundamental operations on this ring.
3. We show that the ring of Witt vectors over $\mathbb{Z}/p\mathbb{Z}$ is isomorphic to $\mathbb{Z}_p$.

Parts 1 and 2 are independent of each other; part 3 bridges the first two.

To give the reader a high-level overview of the mathematical content of our formalization, we sketch here the route that we will follow through these parts. Since there is extensive introductory literature on the p-adic numbers we focus on the latter parts. Our main reference for part 1 is Gouvêa [11], although much is folklore. Parts 2 and 3 primarily follow Hazewinkel [14].

2.1 $\mathbb{Q}_p$ and $\mathbb{Z}_p$

The analytic perspective on the *p-adic numbers* $\mathbb{Q}_p$ defines them analogously to the real numbers $\mathbb{R}$. For a fixed prime number p, $\mathbb{Q}_p$ is the Cauchy completion of the rationals $\mathbb{Q}$ with respect to the p-adic norm, an alternative to the familiar absolute value which is small for numbers whose numerators are divisible by large powers of p. The field operations and norm on $\mathbb{Q}$ lift to $\mathbb{Q}_p$. The *p-adic integers* $\mathbb{Z}_p$ are the p-adic numbers with norm at most 1; they form a ring.

We can alternatively give an algebraic characterization of the p-adics. From this perspective, we take $\mathbb{Z}_p$ to be the projective limit of $\mathbb{Z}/p^n\mathbb{Z}$ in the category of rings and $\mathbb{Q}_p$ to be the field of fractions of $\mathbb{Z}_p$.

Either perspective allows us to see $z \in \mathbb{Z}_p$ as an infinite sum $\sum_{k=0}^{\infty} z_k p^k$ where $0 \le z_k < p$ for each k. (While this sum diverges in the standard absolute value, it converges in the p-adic norm.) This is particularly clear from the algebraic perspective, as the nth partial sum corresponds to an approximation to z in $\mathbb{Z}/p^n\mathbb{Z}$. One can thus picture a p-adic integer as a left-infinite base-p expansion of digits (Fig. 1).

The p-adic numbers are fundamental of many areas of number theory. Among many other applications, they appear in the studies of Diophantine equations [16] and rational points on algebraic varieties [22], and lie at the core of the Hasse principle in Diophantine geometry [2].

$$\begin{array}{r} 1\,1\,1\,1\,1\,1\,1 \\ \ldots 44444444 \\ + \quad\quad\quad 1 \\ \hline \quad\quad\quad 0 \end{array} \qquad \begin{array}{r} 1\,2\,1\,2\,1\,2\,1 \\ \ldots 31313132 \\ \times \quad\quad\quad 3 \\ \hline \quad\quad\quad 1 \end{array} \qquad \begin{array}{r} 1\,1\,1\,1\,1\,1\,1 \\ \ldots 31313132 \\ + \ldots 44444444 \\ \hline \ldots 31313131 \end{array}$$

Figure 1. If we represent $\mathbb{Z}_p$ as left-infinite streams of digits, we can perform addition and multiplication in base p by carrying remainders to the left. 5-adically, $\ldots 444444 + 1 = 0$ and $\ldots 313132 \times 3 = 1$.

2.2 The Ring of p-typical Witt Vectors

Fix a prime number p and a commutative ring R. The underlying set of the *ring of p-typical Witt vectors* $\mathbb{W}R$ is the set of functions $\mathbb{N} \to R$. (Note that the prime number p is usually suppressed in the notation $\mathbb{W}R$.) One usually pictures a Witt vector x as a left-infinite sequence of *coefficients*:

$$(\ldots, x_i, \ldots, x_2, x_1, x_0), \quad x_i \in R.$$

A very illustrative example to keep in mind is $\mathbb{W}(\mathbb{Z}/p\mathbb{Z})$, in which the coefficients x_i are integers modulo p. Readers may recognize the similarity to $\mathbb{Z}_p$, and we will eventually show that they are isomorphic as rings.

We will now describe some properties of $\mathbb{W}R$.

First, $\mathbb{W}R$ is a commutative ring of characteristic 0, even if R has characteristic $p > 1$. For Witt vectors $x = (\ldots, x_1, x_0)$ and $y = (\ldots, y_1, y_0)$ in $\mathbb{W}R$, the addition and multiplication are defined as follows:

$$\begin{aligned} x + y &= (\ldots, S_i(x,y), \ldots, S_1(x,y), S_0(x,y)) \\ x \cdot y &= (\ldots, P_i(x,y), \ldots, P_1(x,y), P_0(x,y)) \end{aligned} \tag{2.2.1}$$

where the $S_i, P_i \in \mathbb{Z}[\ldots, X_1, X_0, \ldots, Y_1, Y_0]$ are certain polynomials that we will specify in Section 4.2. Importantly, these operations are not the familiar componentwise addition and multiplication of sequences. The nth entry, e.g. $S_n(x,y)$, will depend on the entries $(x_n, \ldots, x_0)$ and $(y_n, \ldots, y_0)$ instead of only x_n and y_n. This is similar to "carrying" arithmetic: an overflow at one index creates a ripple that can reach arbitrarily far to the left. It takes some machinery to establish that these operations satisfy the axioms of a ring.

Second, $\mathbb{W}$ is functorial: for every ring homomorphism $f\colon R \to S$, there is an induced ring homomorphism $\mathbb{W}f\colon \mathbb{W}R \to \mathbb{W}S$ obtained by applying f to all coefficients of x. This procedure preserves identity morphisms and compositions.

Third, we introduce the rings of truncated Witt vectors. For a given natural number n, one may truncate Witt vectors to their first n coefficients, which is compatible with the ring structure. We therefore obtain a ring structure on $\mathbb{W}_nR = R^n$ and ring homomorphisms

$$\mathbb{W}R \to \mathbb{W}_nR, \quad x \mapsto (x_{n-1}, \ldots, x_1, x_0).$$

It is clear from this description that $\mathbb{W}R$ is the projective limit of the rings $\mathbb{W}_nR$. We describe this in more detail in Section 6.1.

Finally, for the purpose of this introduction, there are several standard operations on Witt vectors which in fact are natural transformations: that is, they behave in the expected way with respect to the functoriality of $\mathbb{W}$.

- The Teichmüller lift is a multiplicative, zero-preserving map
$$\tau\colon R \to \mathbb{W}R, \quad r \mapsto (\dots, 0, 0, r).$$
In the example $\mathbb{W}(\mathbb{Z}/p\mathbb{Z}) \cong \mathbb{Z}_p$, the elements $\tau(r) \in \mathbb{Z}_p$ correspond to the $(p-1)$th roots of unity in $\mathbb{Z}_p$ that can be obtained from $\mathbb{Z}/p\mathbb{Z}$ via Hensel's lemma (together with $\tau(0) = 0$).
- Verschiebung ("shift") is an additive map
$$V\colon \mathbb{W}R \to \mathbb{W}R, \quad x \mapsto (\dots, x_2, x_1, x_0, 0).$$
- Frobenius is a ring homomorphism
$$F\colon \mathbb{W}R \to \mathbb{W}R$$
that is defined for general rings R in a somewhat convoluted way. Suffice it to say that if R is a ring of characteristic p, then $f\colon R \to R, r \mapsto r^p$ is a ring homomorphism (also called Frobenius), and in this case $F = \mathbb{W}f$.
- Multiplication by n is denoted
$$[n]\colon \mathbb{W}R \to \mathbb{W}R, \quad x \mapsto n \cdot x,$$
and is an additive map.

These operations satisfy various identities that we discuss in Section 5.3.

2.3 Universal Calculations

In the preceding section we have claimed various identities of a ring-theoretic nature, for example that addition and multiplication on the Witt vectors are commutative and associative, that the Teichmüller lifts are multiplicative, and that Verschiebung is additive.

Generally speaking, there are two strategies that can be applied for the proofs of these relations. From a high-brow perspective, these strategies amount to the same thing, but they are very different from the point of view of implementation. We apply both strategies in our formalization.

Before explaining these strategies, we lay some groundwork that both have in common. For $n \in \mathbb{N}$, the nth *Witt polynomial* is
$$W_n = \sum_{i=0}^{n} p^i \cdot X_i^{p^{n-i}} \in \mathbb{Z}[\dots, X_1, X_0].$$
(The Witt polynomials play a role in defining S_i and P_i in 2.2.1.) If $x = (\dots, x_1, x_0) \in \mathbb{W}R$ is a Witt vector, then $W_n(x) \in R$ is called the nth *ghost component* of x. By definition of the ring structure on $\mathbb{W}R$, this gives a ring homomorphism
$$w_n\colon \mathbb{W}R \to R, x \mapsto W_n(x).$$
These ghost components assemble into a ring homomorphism called the *ghost map*
$$w\colon \mathbb{W}R \to R^{\mathbb{N}}, \quad x \mapsto (w_0(x), w_1(x), \dots),$$
where the ring structure on the codomain is given by pointwise addition and multiplication. The ghost map is not injective in general, but if p is invertible in R, then it is an isomorphism.

Strategy 1.

1. First prove the identity for rings R in which p is invertible. Use the fact that $\mathbb{W}R$ is isomorphic to $R^{\mathbb{N}}$ via the ghost map.
2. Then prove the identity for polynomial rings over the integers: $R = \mathbb{Z}[(X_i)_{i\in I}]$. Use that these rings inject into $\mathbb{Q}[(X_i)_{i\in I}]$, and apply the preceding point.
3. Finally, use the natural surjective ring homomorphism
$$\mathbb{Z}[(X_r)_{r\in R}] \to R, \qquad X_r \mapsto r$$
to deduce the identity for arbitrary rings R.

Strategy 2 (sketch).

1. Ignore the fact that the ghost map is not injective in general.
2. Apply the ghost map to both sides of the identity, and prove that the resulting claim is true in $R^{\mathbb{N}}$.

Hazewinkel [14, p.14, footnote 14] writes of this strategy:

> There are pitfalls in calculating with ghost components as is done here. Such a calculation gives a valid proof of an identity or something else only if it is a universal calculation; that is, makes no use of any properties beyond those that follow from the axioms for a unital commutative ring only.

Strategy 2 is enticing, but it takes careful planning to make it amenable to formalization. We discuss how we have done this in Section 4.4, sidestepping the pitfalls that Hazewinkel warns about. This strategy is a powerful method for formally checking identities between the functions mentioned in Section 2.2: with a simple Lean tactic for performing specific rewrites, typical proofs take only two or three lines of code.

2.4 Witt Vectors over $\mathbb{Z}/p\mathbb{Z}$

We mentioned in Section 2.2 that $\mathbb{W}(\mathbb{Z}/p\mathbb{Z})$ is isomorphic to $\mathbb{Z}_p$. This isomorphism is constructed in the following manner. The ring $\mathbb{W}(\mathbb{Z}/p\mathbb{Z})$ is the projective limit of the rings of truncated Witt vectors $\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z})$. Similarly, $\mathbb{Z}_p$ is the projective limit of the rings $\mathbb{Z}/p^n\mathbb{Z}$. It therefore suffices to construct isomorphisms $\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z}) \to \mathbb{Z}/p^n\mathbb{Z}$ that commute with the natural homomorphisms
$$\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z}) \to \mathbb{W}_m(\mathbb{Z}/p\mathbb{Z}) \quad \text{and} \quad \mathbb{Z}/p^n\mathbb{Z} \to \mathbb{Z}/p^m\mathbb{Z}$$
for all $m \le n$. Since any two morphisms out of $\mathbb{Z}/k\mathbb{Z}$ are always equal, this commutativity condition is vacuously satisfied, and we are left with constructing the isomorphisms

$\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z}) \to \mathbb{Z}/p^n\mathbb{Z}$. Using the fact that $\mathbb{Z}/p\mathbb{Z}$ has characteristic p, one can show that

$$p^i = (\dots, 0, 1, \underbrace{0, \dots, 0}_{i \text{ times}}) \in \mathbb{W}(\mathbb{Z}/p\mathbb{Z}).$$

This means that for all $i < n$ we find $p^i \neq 0$ in $\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z})$. Hence $\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z})$ is a ring of characteristic p^n that has cardinality p^n. It is therefore isomorphic to $\mathbb{Z}/p^n\mathbb{Z}$. This completes the proof that $\mathbb{W}(\mathbb{Z}/p\mathbb{Z})$ is isomorphic to $\mathbb{Z}_p$.

2.5 Lean and mathlib

Our formalization is based on Lean's community-driven mathematical library mathlib [21]. It is in some ways difficult to draw a line between our development and mathlib: many results implemented for this project have already entered the library and we expect to incorporate the rest over time. We depend on numerous modules in the library that have been enhanced by earlier projects. In particular, Lewis' construction of $\mathbb{Z}_p$ [17] and preliminaries from Buzzard, Commelin, and Massot's work on the theory of valuation rings [3] serve as a solid foundation for our work.

We rely heavily on the theory of multivariate polynomials, to which many community members have contributed. The type `mv_polynomial σ R`, where `R` is a commutative semiring, represents polynomials with coefficients in `R` whose variables are indexed by the type σ.

Lean's core library and mathlib are designed around using *type classes* [27, 29] to manage mathematical structure. Our development takes this path as well. Structures in mathlib usually follow a *partially bundled* approach, where, for example, `group G` is a `Type`-valued predicate on a type `G` asserting that `G` has a group structure. While the group operations and their properties are bundled in the structure definition, the carrier type `G` is not.

An exception to this rule is mathlib's use of bundled morphisms [21, Section 4.1.2]. The partially bundled approach would suggest to define a type class `is_ring_hom f` asserting that `f : R → S` satisfies the properties of a ring homomorphism. (The ring structures on `R` and `S` are provided by type class arguments.) In practice, the issues with compositionality introduced by this approach are worse than the problems it solves. Instead, mathlib defines a structure `ring_hom R S`, with notation `R →+* S`, that bundles a function `R → S` with proofs that it satisfies the ring homomorphism properties. A coercion from `R →+* S` to `R → S` projecting out this function allows us to apply ring homomorphisms as if they were native functions. While at first glance this may seem to cut against the grain of the type theory, in practice it works without issue, and behaves predictably in its interactions with Lean's type class inference and simplifier. We use the same approach for ring isomorphisms `R ≃+* S`.

Some Lean code snippets in this paper have been slightly edited for the sake of formatting. We fix parameters `p : ℕ` and `R : Type` throughout, assuming `p` is prime and `R` is a commutative ring.

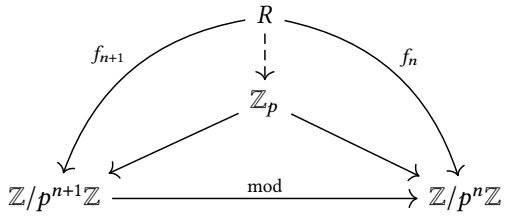

Figure 2. $\mathbb{Z}_p$ is the projective limit of $\mathbb{Z}/p^n\mathbb{Z}$. Any family of compatible morphisms $f_n \colon R \to \mathbb{Z}/p^n\mathbb{Z}$ factors uniquely through $\mathbb{Z}_p$.

3 Algebra of $\mathbb{Z}_p$

We begin with the mathlib development of the p-adic numbers described by Lewis [17]. This development defines $\mathbb{Q}_p$ as the Cauchy completion of $\mathbb{Q}$ with respect to the p-adic norm and $\mathbb{Z}_p$ as the subring of elements with norm at most 1. It establishes some basic algebraic facts about $\mathbb{Z}_p$, including that it is a local ring with maximal ideal spanned by p. Our goal is to further develop the algebraic theory of $\mathbb{Z}_p$, culminating in a proof of its universal property (Fig. 2), that it is the projective limit of the rings $\mathbb{Z}/p^n\mathbb{Z}$.

We follow mathlib in using the notation `ℤ_[p]` for the Lean type `padic_int p`.

3.1 Algebraic Instances

We first establish that $\mathbb{Z}_p$ is a discrete valuation ring (DVR). We will need to know something about the structure of the ideals of $\mathbb{Z}_p$.

In the interest of developing a full API, we prove a number of lemmas characterizing open unit balls. These are mostly variants of the following:

```
lemma norm_le_pow_iff_mem_span_pow
  (x : ℤ_[p]) (n : ℕ) :
  ‖x‖ ≤ p ^ (-n : ℤ) ↔
    x ∈ (ideal.span {p ^ n} : ideal ℤ_[p])
```

In addition to lifting the p-adic norm from $\mathbb{Q}$ to $\mathbb{Q}_p$ and $\mathbb{Z}_p$, it is also useful to lift the p-adic valuation v_p. This was done by Buzzard, Commelin, and Massot [3] but not integrated into mathlib; we integrate their work and provide variants in terms of this valuation, e.g.

```
lemma mem_span_pow_iff_le_valuation
  {x : ℤ_[p]} (hx : x ≠ 0) (n : ℕ) :
  x ∈ (ideal.span {p ^ n} : ideal ℤ_[p]) ↔
    ↑n ≤ x.valuation
```

Proving these results is straightforward. The `norm_cast` tactic [18], developed to simplify expressions containing type coercions, proved useful to handle the many embeddings between $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}_p$, and $\mathbb{Z}_p$.

These various characterizations of the ideals of $\mathbb{Z}_p$ and the fact that p is prime in $\mathbb{Z}_p$ are sufficient to show that $\mathbb{Z}_p$ is a DVR. Unfortunately DVRs provide an excellent example of a familiar pitfall of formalization. Wikipedia provides 10 equivalent characterizations of a DVR, each one convenient in certain contexts, but in a proof assistant we must choose one as primary. We found that the existing mathlib definition was not well suited to our application, and had to develop an alternate characterization and prove it equivalent to the existing criterion.

3.2 Universal Property

One can think of an element of $\mathbb{Z}_p$ as a left-infinite base-p expansion of numerals. With this in mind, it is possible to visualize a map from $\mathbb{Z}_p$ to $\mathbb{Z}/p^k\mathbb{Z}$ for $k \in \mathbb{N}$: take the k rightmost digits of the expansion. It is perhaps harder to see how to define this on the analytic representation of $\mathbb{Z}_p$ or that this operation is a ring homomorphism.

We define this family of homomorphisms recursively, first handling the $k = 1$ case and then using this in the general case. The definitions are similar, so we factor out a common constructor: to produce a ring homomorphism `ℤ_[p] →+* zmod k`, it suffices to give `f : ℤ_[p] → ℕ` satisfying certain properties. Here, `zmod k` is the mathlib representation of $\mathbb{Z}/k\mathbb{Z}$, the ring of integers modulo `k`.

```
def to_zmod_hom (k : ℕ) (f : ℤ_[p] → ℕ)
  (f_spec : ∀ x,
    x - f x ∈ (ideal.span {k} : ideal ℤ_[p]))
  (f_congr : ∀ (x : ℤ_[p]) (a b : ℕ),
      x - a ∈ (ideal.span {k} : ideal ℤ_[p]) →
        x - b ∈ (ideal.span {k} : ideal ℤ_[p]) →
          (a : zmod k) = b) :
  ℤ_[p] →+* zmod k
```

Suppose $r \in \mathbb{Q}$ with $\|r\|_p \leq 1$. There is a unique integer $0 \leq m(p,r) < p$ such that $\|r - m(p,r)\|_p < 1$. Using that $\mathbb{Q}$ is densely embedded in $\mathbb{Q}_p$, we can transfer this property from $\mathbb{Q}$ to $\mathbb{Q}_p$, and rephrase using results from Section 3.1 as the lemma:

```
lemma exists_mem_range (x : ℤ_[p]) :
  ∃ n : ℕ, n < p ∧ (x - n ∈ maximal_ideal ℤ_[p])
```

The function `zmod_repr : ℤ_[p] → ℕ` projects out this value `n`. By construction, it satisfies the `f_spec` requirement of `to_zmod_hom`, and after a little more work to establish `f_congr` we can define `to_zmod : ℤ_[p] →+* zmod p`.

For the general case, we must define a family of functions `appr : ℤ_[p] → ℕ → ℕ` such that `appr x n` satisfies `f_spec x` and `f_congr x` for `k = p^n`. These are effectively the "n rightmost digits" functions mentioned above, approximating `x` to `n` places.

The key to defining `appr x` is to note that, for $x \neq 0$, there is a unique unit element $u \in \mathbb{Z}_p$ such that $x = u \cdot p^{|v_p(x)|}$. We call this element `unit_coeff x`. We then define `appr x n` by recursion on `n : ℕ`.

```
def appr : ℤ_[p] → ℕ → ℕ
| x 0     := 0
| x (n+1) :=
let y := x - appr x n in
if hy : y = 0 then appr x n
else let u := unit_coeff hy,
         v := |y.valuation - n|,
         d := to_zmod (u * (p ^ v)) in
  appr x n + p ^ n * d.val
```

In the recursive case, we take `y` to be the error in the previous approximation, and apply `to_zmod` to a product of `unit_coeff y`. This is the $(n+1)$th rightmost digit of our expansion, so we can scale it and add it to the previous approximation. After proving the specification and congruence properties of `appr`, we again use `to_zmod_hom` to define:

```
to_zmod_pow (n : ℕ) : ℤ_[p] →+* zmod (p ^ n)
```

The construction of `appr` may sound like a complicated way to define a function with an intuitively simple description, and indeed it takes some work to establish `f_spec` and `f_congr`. It would be drastically simplified if we began with an algebraic definition of $\mathbb{Z}_p$ instead of the analytic one. However, the complexity might resurface in other places.

These analytic results are on display in the final step of this section, when we show that $\mathbb{Z}_p$ is the projective limit of $\mathbb{Z}/p^n\mathbb{Z}$. For a fixed ring R, we work with a family of ring homomorphisms $f_k : R \to \mathbb{Z}/p^k\mathbb{Z}$ which we assume to be *compatible*: for any r and $k_1 \leq k_2$, $f_{k_1}(r) \equiv f_{k_2}(r) \mod p^{k_1}$. For any r, the sequence $n \mapsto f_n(r) \in \mathbb{Z}$ is Cauchy in the p-adic norm, and thus converges in $\mathbb{Z}_p$. Calculations establish that this map $R \to \mathbb{Z}_p$ is a ring hom, so we can define:

```
def lift (f : Π (k : ℕ), R →+* zmod (p ^ k))
  (f_compat : ...) : R →+* ℤ_[p]
```

We finally show that `lift` is the unique function satisfying the commutative diagram in Fig. 2, establishing the universal property of $\mathbb{Z}_p$ as the projective limit of $\mathbb{Z}/p^n\mathbb{Z}$.

This result will be essential in Section 6. There, we will prove that $\mathbb{W}(\mathbb{Z}/p\mathbb{Z})$ satisfies the same universal property, and conclude that the two rings are isomorphic. In the meantime we face the substantial task of defining $\mathbb{W}$ and its ring structure.

4 Witt Polynomials and Vectors

We can now continue to work toward the definition of $\mathbb{W}$. While the bare definition is very easy to state, we will need some machinery to define its ring structure, so we develop that machinery first.

4.1 Monadic Approach to Polynomials

Key to simplifying statements in the realm of universal calculations is the monadic `bind` operation on the type of polynomials. We often need to evaluate polynomials on other polynomials, and defining it (together with a good collection

of simplification lemmas) made many calculations straightforward.

Given `f : σ → mv_polynomial τ R`, we define an algebra homomorphism

`bind₁ f : mv_polynomial σ R →ₐ[R] mv_polynomial τ R`

that evaluates a polynomial in variables of type σ by sending each variable to its image under `f`. The subscript $_1$ distinguishes this from an analogous operation that acts on the coefficient ring instead of the variables, but we do not use $\texttt{bind}_2$ in our current development.

The $\texttt{bind}_1$ operator appears in many of our definitions and specifications, and interacts naturally with the various Witt vector operations. We register these interactions as simplification lemmas. One can think of $\texttt{bind}_1$ as an atom in the universal language of rings: when calculating, the definition $\texttt{bind}_1$ should never be unfolded, and once other definitions are unfolded to the $\texttt{bind}_1$ level the simplifier can often finish the calculation.

Note that in the informal notation, this operation is transparent, and hence the calculations, involving say associativity of $\texttt{bind}_1$ and renaming of variables, don't need to be performed either. For our informal presentation here we will denote the function $\texttt{bind}_1$ `f` by bind_f.

The `bind` operator does indeed induce a lawful monad structure on `mv_polynomial`. Its corresponding `pure` operator is the polynomial variable operator

`X : σ → mv_polynomial σ R`

which lifts a term of the variable index type σ to a polynomial. Its map operator, `rename`, reindexes the variables via a map $\sigma \to \tau$.

4.2 Witt Polynomials and Structure Polynomials

We can now define the Witt polynomials, which we will use to describe the ring structure on $\mathbb{W}R$. Recall that for $n \in \mathbb{N}$, the nth *Witt polynomial* is

$$W_n = \sum_{i=0}^{n} p^i \cdot X_i^{p^{n-i}} \in \mathbb{Z}[\dots, X_1, X_0]. \tag{4.2.1}$$

Their Lean definition is a direct translation of Eq. (4.2.1):

```
def witt_polynomial (n : ℕ) : mv_polynomial ℕ R :=
Σ i in range (n+1),
  monomial (single i (p ^ (n - i))) (p ^ i)
```

We use the notation `W_ R n` for this type.

It is not so hard to see that over the rationals, but not the integers, the polynomials W_n form an alternative basis of the polynomial algebra $\mathbb{Q}[\dots, X_1, X_0]$, so that by abuse of notation we may write

$$\mathbb{Q}[\dots, W_1, W_0] \cong \mathbb{Q}[\dots, X_1, X_0].$$

In Lean, we define polynomials `X_in_terms_of_W p R n` that correspond to `X n` viewed on the basis of Witt polynomials. In other words, applying $\texttt{bind}_1$ `(W_ R)` to the polynomial `X_in_terms_of_W p R n` produces `X n`, and similarly if we swap the polynomials. This fact is key to establishing the algebra automorphism that makes it easy to prove the following lemma. For reasons of exposition, we only treat the case where Φ is a polynomial in two variables, but apart from notational complexity the case of an arbitrary (even infinite) number of variables is not different at all.

Lemma 4.2.2. *Let $\Phi \in \mathbb{Q}[X, Y]$ be a polynomial. Then there exists a unique sequence of polynomials*

$$\phi_n \in \mathbb{Q}[\dots, X_1, X_0, \dots Y_1, Y_0], \quad (n \in \mathbb{N})$$

such that for all natural numbers n

$$W_n(\dots \phi_1, \phi_0) = \Phi(W_n, W_n).$$

The monadic $\texttt{bind}_1$ makes another appearance in the formal statement of this lemma:

```
theorem witt_structure_rat_exists_unique
  (Φ : mv_polynomial idx ℚ) :
∃! (φ : ℕ → mv_polynomial (idx × ℕ) ℚ),
  ∀ (n : ℕ), bind₁ φ (W_ ℚ n) =
    bind₁ (λ i, (rename (prod.mk i) (W_ ℚ n))) Φ
```

A non-trivial calculation shows that if Φ has integral coefficients, then so do the ϕ_n. Thus we get the following key theorem, on which the whole theory of Witt vectors relies.

Theorem 4.2.3. *Let $\Phi \in \mathbb{Z}[X, Y]$ be a polynomial. Then there exists a unique sequence of polynomials*

$$\phi_n \in \mathbb{Z}[\dots, X_1, X_0, \dots Y_1, Y_0], \quad (n \in \mathbb{N})$$

such that for all natural numbers n

$$W_n(\dots \phi_1, \phi_0) = \Phi(W_n, W_n).$$

The details of implementing this non-trivial calculation are not pleasant, involving arguments about the badly behaved numerator and denominator functions. This is indeed one of the few points at which we step outside the language of rings. The key ingredient in the proof is the following basic but non-trivial number-theoretic fact.

```
lemma dvd_sub_pow_of_dvd_sub {p : ℕ} {a b : R}
  (h : (p : R) ∣ a - b) (k : ℕ) :
  (p^(k+1) : R) ∣ a^(p^k) - b^(p^k)
```

Coq's Mathematical Components library [20] provides an interface for manipulating polynomials whose coefficients lie in a subring of a base ring. There is no analogous interface for mathlib's multivariate polynomials, but in retrospect, it seems likely that this approach, with base ring $\mathbb{Q}$ and subring $\mathbb{Z}$, may have helped here.

The sequences of polynomials S_n and P_n that occur in the definition (Eq. (2.2.1)) of the addition and multiplication on $\mathbb{W}R$ will be obtained by applying this theorem to the polynomials $X + Y$ and $X \cdot Y$ respectively. We explain in Section 5.1 why these operations satisfy the axioms of a commutative ring.

4.3 The Type of p-adic Witt Vectors

After Section 4.4, we will have all the necessary machinery to define a ring structure and operations on $\mathbb{W}R$. Before that, though, we must specify what a Witt vector actually is.

This part of the definition is fortunately easy. As indicated in Section 2.2, a Witt vector over R is an infinite stream of coefficients in R,

$$(\ldots, x_i, \ldots, x_2, x_1, x_0), \quad x_i \in R.$$

This leads to perhaps the simplest definition in our formalization:

```
def witt_vector (p : ℕ) (R : Type*) := ℕ → R
```

The argument `p` is not used in the definition, but `witt_vector p R` will have a different ring structure for each `p`.

4.4 Universal Calculations

In Section 2.3, we sketched a strategy for proving identities between operators on the ring of Witt vectors. This strategy was imprecise, and as Hazewinkel wrote, it only gives a valid proof if it is a "universal calculation; that is, makes no use of any properties beyond those that follow from the axioms for a unital commutative ring only."

In the remainder of this paper, we will use the term "universal calculation" in the following precise way: it is a calculation with *polynomial functions* on the ring of Witt vectors. Let us now explain what we mean by a polynomial function.

Many of the operations on $\mathbb{W}R$ that we will study have a polynomial structure to them. Let $f : \mathbb{W}R \to \mathbb{W}R$ be such an operator. We say that f is a *polynomial function* if there is a family of polynomials $\varphi_n \in \mathbb{Z}[X_0, X_1, \ldots]$ such that for each $n \in \mathbb{N}$ and $x = (\ldots x_1, x_0) \in \mathbb{W}R$,

$$f(x)_n = \varphi_n(x_0, x_1, \ldots).$$

We formalize this as a `Prop`-valued relation between the function and the family of polynomials.

```
def is_poly
  (f : Π {R} [comm_ring R], 𝕎 R → 𝕎 R)
  (φ : ℕ → mv_polynomial ℕ ℤ) : Prop :=
∀ {R} [comm_ring R] (x : 𝕎 R),
  (f x).coeff = λ n, aeval x.coeff (φ n)
```

The power of this predicate comes from its extensionality principle, a corollary of Theorem 4.2.3.

Lemma 4.4.1. *Let $f, g : \mathbb{W}R \to \mathbb{W}R$, and let $\phi_n, \psi_n \in \mathbb{Z}[\ldots, X_1, X_0]$ be two families of polynomials indexed by $\mathbb{N}$ witnessing that f and g, respectively, are polynomial functions. If for all $n \in \mathbb{N}$ we have*

$$W_n(\ldots, \phi_1, \phi_0) = W_n(\ldots, \psi_1, \psi_0),$$

then $\phi_n = \psi_n$ for all $n \in \mathbb{N}$, and hence $f = g$.

In other words, two polynomial functions f and g are equal when we obtain identical values when evaluating the Witt polynomials on their underlying polynomials. Note that the condition

$$W_n(\ldots, \phi_1, \phi_0) = W_n(\ldots, \psi_1, \psi_0)$$

can be written equivalently as

$$\mathrm{bind}_\varphi(W_n) = \mathrm{bind}_\psi(W_n).$$

Even though this condition may look intimidating, it is very pleasant to check in practice, as it often reduces to an expression in a restricted language that can be proved by the simplifier (Fig. 3). The "predictable rewriting" proofs we will discuss in Section 5.2 typically follow applications of this extensionality principle. Since the composition of polynomial functions is polynomial, we only need to establish a small number of atomic cases in order to apply this principle widely.

For example, we apply this strategy to check the relation $F \circ V = [p]$. The proof in Lean is simply:

```
lemma frobenius_verschiebung (x : 𝕎 R) :
  frobenius (verschiebung x) = x * p :=
is_poly.ext'
  ((frobenius_is_poly p).comp
    verschiebung_is_poly)
  (mul_n_is_poly p p)
  (by witt_simp) _ _
```

What we have just described is, in fact, the precise version of the second strategy for calculating with ghost components that we sketch in Section 2.3. The restricted language that our technique targets is that of unital commutative rings and morphisms between them. Essentially, the calculations we carry out in this language may not depend on features of the specific rings in question: they may not assume that

```
∀ (n : ℕ),
 ⇑(bind₁ (λ (n : ℕ),
  ⇑(bind₁ (λ (n : ℕ),
   ⇑(bind₁ (function.uncurry
    ![λ (k : ℕ), ⇑(rename (prod.mk 0)) (X k),
     λ (k : ℕ), ⇑(rename (prod.mk 1)) (frobenius_poly p k)]))
   (witt_mul p n)))
  (verschiebung_poly n)))
 (witt_polynomial p ℤ n) =
  ⇑(bind₁ (λ (n : ℕ),
   ⇑(bind₁ (function.uncurry
    ![λ (k : ℕ), ⇑(rename (prod.mk 0)) (verschiebung_poly k),
     λ (k : ℕ), ⇑(rename (prod.mk 1)) (X k)]))
    (witt_mul p n)))
   (witt_polynomial p ℤ n)
```

Figure 3. The goals produced by Lemma 4.4.1 typically have a structure similar to this, which appears in the proof of an identity from Section 5.3. With the $\mathtt{bind}_1$ abstraction, the simplifier can feasibly unfold the polynomial structures and normalize both sides of the equation without any direction from the user.

p is invertible, that the rings have certain characteristic, or anything of the sort.

We can thus rephrase our strategy from before:

Strategy 2.

- Show that both sides of the identity are given by $\mathbb{N}$-indexed families of polynomial operations on the coefficients of Witt vectors.
- Show that these polynomial operations are equal.
- Use Lemma 4.4.1 to reduce this to a computation on ghost components.

The notion of a polynomial function $\mathbb{W}R \to \mathbb{W}R$ extends, in an obvious way, to functions $(\mathbb{W}R)^n \to \mathbb{W}R$ of any arity. Addition and multiplication of Witt vectors, for instance, are polynomial by definition. Defining binary versions of the predicate, extensionality lemma, and composition rules further increase the opportunities to use this strategy.

Strategy 2 is powerful and straightforward. The downside is that it is hard to turn the principle into a fully generic and flexible machine. One limitation appears when we consider τ (Teichmüller), a function $R \to \mathbb{W}R$, which doesn't fit in the framework of n-ary functions from $\mathbb{W}R$ to itself. The lack of a convenient library in mathlib for the composition of n-ary functions prevents us from using an `is_poly` predicate on functions of arbitrary arity. For our current applications, this is no great barrier. We are able to use Strategy 1 to work around these restrictions when needed. In the future, it would be interesting to extend our technique to make this strategy more widely applicable.

5 Ring Structure and Other Operations

Our task now is to define the ring structure on $\mathbb{W}R$ and the Teichmüller, Verschiebung, Frobenius, and multiplication-by-n operations. The Teichmüller operator will not make an appearance in Section 6, but we include it in the interest of establishing a general interface for Witt vectors in mathlib.

Our proofs proceed, as much as possible, as universal calculations. Following Strategy 2 as explained in the previous section allows many proofs to use essentially the same arguments, for instance, when we establish the homomorphism properties of the operators. These arguments are similar enough that we were able to factor them into short metaprograms, only slightly more complicated than tactic macros, that can replicate them with minimal user input.

5.1 The Ring of Witt Vectors

In Section 2.2 we defined, for Witt vectors $x = (\dots, x_1, x_0)$ and $y = (\dots, y_1, y_0)$ in $\mathbb{W}R$,

$$x + y = (\dots, S_i(x, y), \dots, S_1(x, y), S_0(x, y))$$
$$x \cdot y = (\dots, P_i(x, y), \dots, P_1(x, y), P_0(x, y))$$

for then-unspecified families of polynomials S_i and P_i. We obtain these families, which we call *structure polynomials*, by applying Theorem 4.2.3 to the bivariate polynomials $X + Y$ and $X \cdot Y$. The structure polynomial for negation is obtained in the same way using the univariate polynomial $-X$.

In Lean, the unique family of polynomials from Theorem 4.2.3 goes by the name `witt_structure_int`. We define:

```
def witt_add :
  ℕ → mv_polynomial (fin 2 × ℕ) ℤ :=
witt_structure_int p (X 0 + X 1)

def witt_mul :
  ℕ → mv_polynomial (fin 2 × ℕ) ℤ :=
witt_structure_int p (X 0 * X 1)

def witt_neg :
  ℕ → mv_polynomial (fin 1 × ℕ) ℤ :=
witt_structure_int p (-X 0)
```

The addition on $\mathbb{W}R$ is then defined by letting the nth coefficient be the evaluation of `witt_add n` on the coefficients of $x, y \in \mathbb{W}R$. Multiplication, negation, and the elements 0 and 1 are defined similarly.

```
def eval {k : ℕ}
  (φ : ℕ → mv_polynomial (fin k × ℕ) ℤ)
  (x : fin k → 𝕎 R) : 𝕎 R :=
mk p $ λ n, peval (φ n) $ λ i, (x i).coeff

instance : has_add (𝕎 R) :=
⟨λ x y, eval (witt_add p) ![x, y]⟩
```

To show that these definitions make $\mathbb{W}R$ into a commutative ring, we must check that they satisfy a number of axioms. Doing this explicitly would be tedious. We therefore follow Strategy 1, as explained in Section 2.3.

Suppose that $f\colon R \to S$ is a function, both R and S are endowed with 0, 1, +, ·, and −, and the function f preserves this structure. If f is injective, and S satisfies the axioms of a commutative ring, then so does R. In Lean this fact is recorded in `function.injective.comm_ring`. Dually, if f is surjective, and R satisfies the axioms of a commutative ring, then so does S. This is `function.surjective.comm_ring`.

For every ring homomorphism $f\colon R \to S$, the map

$$\mathbb{W}f\colon \mathbb{W}R \to \mathbb{W}S, \quad (\dots, x_1, x_0) \mapsto (\dots, f(x_1), f(x_0))$$

preserves the ring operations. We prove this in five lemmas, one lemma for each of 0, 1, +, ·, and −. These goals are uniform enough that all can be proved by the same five line tactic script, which we factor into a tactic macro. This is not the only place we encounter repetitive goals like this, and we elaborate on the use of auxiliary tactics in Section 5.2.

We can then argue as follows that $\mathbb{W}R$ is a commutative ring:

- Recall from Section 2.3 the ring homomorphism
  $$w\colon \mathbb{W}R \to R^{\mathbb{N}}, \quad x \mapsto (W_0(x), W_1(x), \dots),$$
  called the ghost map. If p is invertible in R, then the ghost map is injective. So in this case $\mathbb{W}R$ is a commutative ring.

- If $R = \mathbb{Z}[(X_i)_{i\in I}]$ is some polynomial algebra over the integers, then we use the natural injection
$$\mathbb{W}(\mathbb{Z}[(X_i)_{i\in I}]) \to \mathbb{W}(\mathbb{Q}[(X_i)_{i\in I}])$$
and the fact that it preserves the ring operations, as discussed above. Since p is invertible in $\mathbb{Q}[(X_i)_{i\in I}]$, we deduce that $\mathbb{W}(\mathbb{Z}[(X_i)_{i\in I}])$ is a commutative ring from the preceding point and `function.injective.comm_ring`.
- Finally, for arbitrary commutative rings R, consider the map $\mathbb{W}f$, where f is the natural surjection
$$f\colon \mathbb{Z}[(X_r)_{r\in R}] \to R.$$
Since f is surjective, so is $\mathbb{W}f$, and we can therefore conclude that $\mathbb{W}R$ is a commutative ring from the fact that $\mathbb{W}(\mathbb{Z}[(X_r)_{r\in R}])$ is a commutative ring.

We have used Strategy 1, as opposed to Strategy 2 which was explained in Section 4.4, for two reasons.

(*i*) With our current approach we deduce all the axioms at once, whereas with Strategy 2 we would have to check them one by one.

(*ii*) The associativity axioms refer to ternary operations, and we have only formalized the machinery of Strategy 2 in the unary and binary setting. So far, we haven't found a direct use for higher arity versions besides the associativity axioms, and without a convenient way to uniformly handle n-ary versions, it was not worth the effort to develop ternary machinery for this single application.

5.2 Auxiliary Tactics

An appealing feature of Lean as a proof assistant is the easy accessibility of its metaprogramming framework [8]. Lean metaprograms are written in an extension of the language of Lean itself. With very little syntactic overhead, these metaprograms can implement tactics ranging from straightforward macros to procedures that interact with the parser and environment in complex ways.

Our adherence to universal calculations leads to a number of proofs that are identical modulo a few key lemmas or parameters. A typical example of this is in the previous section, when we show that the ghost maps respect the ring operations. The creative step of these proofs is to provide an input polynomial and the correct arguments for use by the Witt structure polynomials; otherwise, the proofs proceed by predictable rewriting.

These predictable proofs fall just outside the scope of a tactic macro, but can be handled easily by a metaprogram that parses and inserts arguments. In the case of the ghost map morphism properties, a metaprogram that proves all four cases is only a few lines longer than a direct proof of one case. We use this approach a number of times while constructing the ring structure on $\mathbb{W}R$. These metaprograms are only used locally, but let us avoid code duplication and highlight the universality of the proof approach.

We are in an even better position now that the ring structure on $\mathbb{W}R$ has been established. A custom set of simplifier lemmas, containing the proper universal ghost component equations, ring homomorphism rules, and some other glue, is able to handle the "predictable rewriting" step across a variety of different applications. In the following section, we will see that properties of V, F, and $[n]$ can all be established in a uniform way, after a small amount of setup, by rewriting with this simp set. Even this small amount of setup is mostly mechanical and could in principle be automated, although the need to handle edge cases means that doing so would not lower the total number of lines of code.

5.3 Operators

Recall from Section 2.2 the four operators Verschiebung (V), Frobenius (F), scalar multiplication ($[n]$), and Teichmüller (τ). For $x, y \in \mathbb{W}R$, these operations satisfy
$$F \circ V = [p]$$
$$V(x \cdot F(y)) = V(x) \cdot y$$
and if R has characteristic p
$$F(x) = (\ldots, x_2^p, x_1^p, x_0^p)$$
$$[p](x) = (\ldots, x_2^p, x_1^p, x_0^p, 0)$$
$$V \circ F = [p]$$
$$p = (\ldots, 0, 0, 1, 0).$$
We will need most of these operations and properties in Section 6.2, although the Teichmüller lift is included only for the sake of completeness. Teichmüller also distinguishes itself as the one operator whose properties we cannot establish via Strategy 2. We will show that each of the others is a polynomial function.

Verschiebung. The definition of the Verschiebung map,
$$V\colon \mathbb{W}R \to \mathbb{W}R, \quad (\ldots, x_2, x_1, x_0) \mapsto (\ldots, x_2, x_1, x_0, 0),$$
translates easily to Lean:

```
def verschiebung_fun (x : 𝕎 R) : 𝕎 R :=
mk p (λ n, if n = 0 then 0 else x.coeff (n - 1))
```

Its underlying polynomial structure is similarly straightforward:

```
def versch_poly (n : ℕ) : mv_polynomial ℕ ℤ :=
if n = 0 then 0 else X (n-1)
```

One lemma, an identity for $\mathrm{bind}_V(W_n)$, is somewhat tedious. Otherwise, it is routine to show that V is indeed a polynomial function, respects addition, is a natural transformation, interacts with the ghost components.

Multiplication by n. For any $n \in \mathbb{N}$, multiplication by n in the ring of Witt vectors
$$[n]\colon \mathbb{W}R \to \mathbb{W}R, \quad x \mapsto n \cdot x$$

is a polynomial function, because it is repeatedly applied addition, which is polynomial. The operation needs no definition in Lean since the coercion `ℕ → 𝕎 R` and multiplication on `𝕎 R` are known: it is simply `λ x, x * n`.

Frobenius. The next operator puts up more of a fight. If R is a ring of characteristic p, then $f\colon R \to R, r \mapsto r^p$ is a ring endomorphism. We use this to obtain an endomorphism $\mathbb{W}f\colon \mathbb{W}R \to \mathbb{W}R$, taking the image of each input coefficient under f (Section 5.1).

We claim that $\mathbb{W}f$ is a polynomial function, which unlocks the toolkit of universal calculations, as described in Section 4.4. In addition, we can use those polynomials to define an endomorphism $F\colon \mathbb{W}R \to \mathbb{W}R$ for arbitrary rings R that agrees with $\mathbb{W}f$ in the case that R has characteristic p. Unfortunately we cannot use the machinery of Theorem 4.2.3 (`witt_structure_int` in Lean) to derive these polynomials. It holds that

$$\mathrm{bind}_F(W_n) = W_{n+1},$$

but to apply Theorem 4.2.3, we need this to be a polynomial expression in $W_0, \dots, W_n$. Since W_{n+1} contains the variable X_{n+1} it cannot be expressed in terms of the earlier Witt polynomials.

This is a very painful off-by-one error. Absent the `witt_structure_int` machinery, we are forced to define the underlying polynomial structure by hand. The proof that it witnesses that F is a polynomial function mimics the argument lifting Lemma 4.2.2 (over $\mathbb{Q}$) to Theorem 4.2.3 (over $\mathbb{Z}$). While the high level approach is similar, the details are different enough that it is not clear how to unify the calculations.

After establishing that F is polynomial, though, we are back in the realm of universal calculations. Further properties of F follow without excess trouble: for instance, if $x = (\dots x_1, x_0) \in \mathbb{W}R$ and R has characteristic p, then

$$(F(x))_n = x_n^p$$

so that F agrees with $\mathbb{W}f$ as promised.

Teichmüller. The signature of the Teichmüller lift τ is not the same as the previous operators, which means we can neither construct it as a polynomial function not reason with universal calculations. Fortunately, its definition

$$\tau\colon R \to \mathbb{W}R, \quad r \mapsto (\dots, 0, 0, r)$$

is easy to translate directly.

```
def teichmuller_fun (r : R) : 𝕎 R
| 0 := r
| (n+1) := 0
```

After establishing that the nth ghost component of $\tau(r)$ is r^{p^n}, it is straightforward to show that τ is multiplicative and zero-preserving.

While τ is not needed in Section 6, it is an essential part of the Witt vector interface, and so we define it for the sake of completeness. It is a multiplicative map inverse to the ring homomorphism

$$w_0\colon \mathbb{W}R \to R, \quad (\dots, x_1, x_0) \mapsto x_0.$$

In Section 6.1 we will see a universal property of $\mathbb{W}$, namely, that it is the projective limit of the rings of truncated Witt vectors. This shows how to build a ring homomorphism *into* $\mathbb{W}R$. Another universal property of $\mathbb{W}$ shows that to build a ring homomorphism *out of* $\mathbb{W}k$, when k is a perfect ring, it suffices to give suitable values at the Teichmüller representatives. We have not formalized this universal property.

Identities. It is when establishing the identities between these operations that the power of Strategy 2 really shines through. The tactic `witt_simp`, a light wrapper around a custom set of simplification lemmas that represent "universal equations," is able to prove these identities as soon as the underlying polynomial structure is exposed. Just as seen in the example in Section 4.4, we can write:

```
lemma verschiebung_mul_frobenius (x y : 𝕎 R) :
  verschiebung (x * frobenius y) =
    verschiebung x * y :=
begin
  apply is_poly₂.ext'
    (verschiebung_is_poly.comp₂
      ((mul_is_poly₂ p).comp_right
        (frobenius_is_poly p)))
    ((mul_is_poly₂ p).comp_left
      verschiebung_is_poly),
  rintro ⟨⟩; witt_simp [mul_assoc]
end
```

We stress that this technique is general and extensible. As the Witt vector library grows, we expect many more proofs to follow this same pattern, with minimal additions to the simp set used by `witt_simp`.

6 Isomorphism with $\mathbb{Z}_p$

So far we have worked with Witt vectors over an arbitrary ring R. As discussed in Section 2.4, when we specialize R to $\mathbb{Z}/p\mathbb{Z}$, we get a ring isomorphic to $\mathbb{Z}_p$. We show this by proving that $\mathbb{W}(\mathbb{Z}/p\mathbb{Z})$ satisfies the same universal property that we established for $\mathbb{Z}_p$ in Section 3.2.

6.1 Truncated Witt Vectors

Just as we approximated elements of $\mathbb{Z}_p$ by truncating all but the rightmost n digits, so we will truncate Witt vectors to their first n elements. We define the truncated Witt vectors $\mathbb{W}_n R$ to be an n-element vector of elements of R:

```
def truncated_witt_vector
  (p : ℕ) (n : ℕ) (R : Type*) : Type :=
fin n → R
```

The parameter `p : ℕ` is unused in this definition but will determine the ring structure on this type. There is a clear map $\mathbb{W}R \to \mathbb{W}_n R$, and a map in the other direction appends

a stream of 0s to the end of a truncated vector. Using these maps, we can lift the ring operations on $\mathbb{W}R$ to $\mathbb{W}_n R$. An auxiliary tactic as described in Section 5.2 establishes that the truncating map respects the ring operations, and since this map is surjective, we can conclude that $\mathbb{W}_n R$ is a ring and the truncating map is a ring hom.

There is another obvious truncation map $\mathbb{W}_m R \to \mathbb{W}_n R$ for $m \leq n$. It should come as no surprise that this map is, again, a ring hom. It additionally composes well with itself and with the full truncating map. The imaginative reader may now see the lower triangle in Fig. 2, with $\mathbb{W}R$ in the middle and $\mathbb{W}_{n+1}R$ and $\mathbb{W}_n R$ on the sides. Indeed, the rest of the diagram follows with no trouble: given a family of compatible maps $S \to \mathbb{W}_n R$, we can produce a unique map $S \to \mathbb{W}R$.

We have said little about the formalization of this section because there is almost nothing to say. No argument in this file takes more than a few lines of code: ring-theoretic machinery and a few simplification lemmas give us everything practically for free. It is surprising, then, that this section contains one of the rare occasions in which we avoid a ring-theoretic definition. The type `truncated_witt_vector p n R` could have been represented as the quotient of `𝕎 R` by the ideal `⟨x : 𝕎 R | ∀ i < n, x.coeff i = 0⟩`. While elegant in principle, this approach made the definition of coefficients of a truncated Witt vector rather annoying, whereas the more direct definition was entirely free of hassle.

6.2 Constructing the Isomorphism

We now know that $\mathbb{Z}_p$ is the projective limit of $\mathbb{Z}/p^n\mathbb{Z}$ and $\mathbb{W}R$ is the projective limit of $\mathbb{W}_n R$. It is finally time to specialize the arbitrary ring R to $\mathbb{Z}/p\mathbb{Z}$. To establish that $\mathbb{W}(\mathbb{Z}/p\mathbb{Z}) \simeq \mathbb{Z}_p$, it suffices by the uniqueness of the projective limit to show that $\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z}) \simeq \mathbb{Z}/p^n\mathbb{Z}$.

It follows immediately that $|\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z})| = p^n$. A general result shows that a ring R with cardinality n and characteristic n must have a unique isomorphism to $\mathbb{Z}/n\mathbb{Z}$, since both unit elements generate the ring as additive group. Showing that $\mathbb{W}_n(\mathbb{Z}/p\mathbb{Z})$ has characteristic n, though, takes some machinery: the proof invokes both the Frobenius and Verschiebung operators and the identity $F \circ V = [p]$. Interestingly, this is the first and only time in this development that we invoke F and V, but developing the theories of these operators seems to be the shortest path to this result.

This isomorphism commutes with the truncation and mod operators (Fig. 4). We then define a family of ring homomorphisms $\mathbb{W}(\mathbb{Z}/p\mathbb{Z}) \to \mathbb{Z}/p^n\mathbb{Z}$ by composing this isomorphism with the truncation map from the previous section. This family is compatible, and thus the universal property of $\mathbb{Z}_p$ lifts it to a homomorphism $\mathbb{W}(\mathbb{Z}/p\mathbb{Z}) \to \mathbb{Z}_p$. Similarly, composing the isomorphism with the homomorphism $\mathbb{Z}_p \to \mathbb{Z}/p^n\mathbb{Z}$ gives a compatible family of homomorphisms $\mathbb{Z}_p \to \mathbb{W}_n(\mathbb{Z}/p\mathbb{Z})$, which the universal property of $\mathbb{W}$ lifts to a homomorphism $\mathbb{Z}_p \to \mathbb{W}(\mathbb{Z}/p\mathbb{Z})$. The uniqueness of the limit, and some straightforward rewriting, let us quickly establish that these maps are inverses, and thus the two rings are isomorphic.

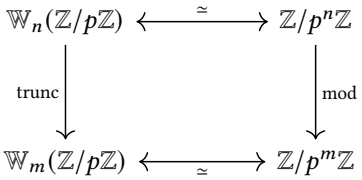

Figure 4. The isomorphism $\mathbb{W}(\mathbb{Z}/p\mathbb{Z}) \simeq \mathbb{Z}/p^n\mathbb{Z}$ commutes with trunc and mod.

```
def equiv : 𝕎 (zmod p) ≃+* ℤ_[p] :=
{ to_fun    := to_padic_int p,
  inv_fun   := from_padic_int p,
  left_inv  := from_padic_comp_to_padic_ext _,
  right_inv := to_padic_comp_from_padic_ext _,
  map_mul   := ring_hom.map_mul _,
  map_add   := ring_hom.map_add _ }
```

7 Concluding Thoughts

The `witt_vector` directory of our mathlib branch contains around 3500 lines of code, including comments and whitespace, discounting preliminaries that will be moved to other locations. Another 1000 lines have been added to the `padics` directory. This counts only material corresponding to sections 3–6. Many thousands more lines of preliminaries, especially about multivariate polynomials, have been or will be incorporated into mathlib. While these comparisons are difficult to make scientifically, we estimate that the 3500 lines correspond to seven dense pages of Hazewinkel [14].

To the best of our knowledge, the ring of Witt vectors has never before been defined in a proof assistant. Lewis [17] surveys the formal developments of p-adic numbers appearing in the literature. While Pelayo, Voevodsky, and Warren [23] take an algebraic approach to defining $\mathbb{Z}_p$ that may be amenable to establishing its advanced algebraic properties, their development does not go beyond the basic ring structure. Other proof assistant libraries defining $\mathbb{Z}_p$ appear to be similarly limited.

Of course, many libraries contain substantial algebraic developments. In particular, Coq's Mathematical Components library [20] contains enough group theory to support Gonthier et al's formalization of the odd order theorem [10]. Others, including Cano et al [4], have enriched the library's ring theory content, but have focused on computational aspects. Isabelle's HOL-Algebra library covers many ring-theoretic topics and an entry by Bordg [1] in the Archive of Formal Proofs constructs ring localizations; the Mizar Mathematical Library also contains a number of articles on ring theory, including by Korniłowicz and Schwarzweller [15] and Watase [30]. Avelar et al [6] describe a formalization of

elementary ring theory in PVS. We are not aware of a development of DVRs or related topics in any of these systems.

Much has been written about different methods for defining and maintaining hierarchies of algebraic structures in proof assistants [12, 19, 24, 27]. In some sense, our project is orthogonal to this literature: we work at a single fixed point within mathlib's type class hierarchy. Nonetheless, there may be some insight here. An early attempt at defining Witt vectors in Lean succumbed to type class searches that were inexplicably long and slow. A combination of library refactoring and improved caching in Lean 3's type class inference have largely resolved these performance issues. Library refactoring, of course, is rarely fun, and it is preferable to design hierarchies right the first time. Tools like Cohen, Sakaguchi, and Tassi's Hierarchy Builder [5] show enormous promise here. The tabled type class resolution procedure implemented in Lean 4 by Selsam, Ullrich, and de Moura [26] will also allow more flexibility in hierarchy design.

While we were not expecting it from the start, a very limited amount of Lean metaprogramming ended up tidying our proof scripts significantly (Section 5.2). These tactics did not just shorten the scripts, but reduced many of them to the point where the human input—expressions and references to lemmas—was essentially the same as it would be informally. We stress that writing these simple tactics requires no knowledge of the proof assistant's architecture or foundations and minimal familiarity with the metaprogramming framework. Mathematical users, especially those who recognize these repetitive proofs in their own developments, would spend their time well gaining this minimal familiarity.

Future work on this topic could go in various directions. Some directions lift extra structure on the base ring R to extra structure on $\mathbb{W}R$. If R is an integral domain of characteristic p, then $\mathbb{W}R$ is an integral domain; if k is a perfect field of characteristic p, then $\mathbb{W}k$ is a DVR. All the ingredients in the definition of p-adic period ring B_{dR} by Fontaine [9] are now available in Lean. Orthogonally, we could define "big" Witt vectors, of which the p-typical Witt vectors described here are a quotient.

Acknowledgments

The first author receives support from the Deutsche Forschungs Gemeinschaft (DFG) under Graduiertenkolleg 1821 *(Cohomological Methods in Geometry)*.

The second author receives support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 713999, Matryoshka) and from the Dutch Research Council (NWO) under the Vidi program (project No. 016.Vidi.189.037, Lean Forward).

References

[1] Anthony Bordg. 2018. The Localization of a Commutative Ring. *Archive of Formal Proofs* (June 2018). http://isa-afp.org/entries/Localization_Ring.html, Formal proof development.

[2] T. D. Browning. 2018. How often does the Hasse principle hold? In *Algebraic geometry: Salt Lake City 2015*. Proc. Sympos. Pure Math., Vol. 97. Amer. Math. Soc., Providence, RI, 89–102.

[3] Kevin Buzzard, Johan Commelin, and Patrick Massot. 2020. Formalising Perfectoid Spaces. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs* (New Orleans, LA, USA) *(CPP 2020)*. Association for Computing Machinery, New York, NY, USA, 299–312. https://doi.org/10.1145/3372885.3373830

[4] Guillaume Cano, Cyril Cohen, Maxime Dénès, Anders Mörtberg, and Vincent Siles. 2016. Formalized linear algebra over elementary divisor rings in Coq. *Logical Methods in Computer Science* 12, 2 (Jun 2016). https://doi.org/10.2168/lmcs-12(2:7)2016

[5] Cyril Cohen, Kazuhiko Sakaguchi, and Enrico Tassi. 2020. Hierarchy Builder: algebraic hierarchies made easy in Coq with Elpi. (Feb. 2020). https://doi.org/10.4230/LIPIcs.CVIT.2016.23

[6] Andréia B. Avelar da Silva, Thaynara Arielly de Lima, and André Luiz Galdino. 2018. Formalizing Ring Theory in PVS. In *Interactive Theorem Proving - 9th International Conference, ITP 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10895)*, Jeremy Avigad and Assia Mahboubi (Eds.). Springer, 40–47. https://doi.org/10.1007/978-3-319-94821-8_3

[7] Sander R. Dahmen, Johannes Hölzl, and Robert Y. Lewis. 2019. Formalizing the Solution to the Cap Set Problem. In *10th International Conference on Interactive Theorem Proving (ITP 2019) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 141)*, John Harrison, John O'Leary, and Andrew Tolmach (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 15:1–15:19. https://doi.org/10.4230/LIPIcs.ITP.2019.15

[8] Gabriel Ebner, Sebastian Ullrich, Jared Roesch, Jeremy Avigad, and Leonardo de Moura. 2017. A metaprogramming framework for formal verification. *PACMPL* 1, ICFP (2017), 34:1–34:29. https://doi.org/10.1145/3110278

[9] Jean-Marc Fontaine. 1994. Le corps des périodes p-adiques. Number 223. 59–111. With an appendix by Pierre Colmez, Périodes p-adiques (Bures-sur-Yvette, 1988).

[10] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. 2013. A Machine-Checked Proof of the Odd Order Theorem. In *ITP 2013*. 163–179. https://doi.org/10.1007/978-3-642-39634-2_14

[11] Fernando Q. Gouvêa. 1997. *p-adic Numbers* (second ed.). Springer, Berlin. vi+298 pages. https://doi.org/10.1007/978-3-642-59058-0

[12] Adam Grabowski, Artur Kornilowicz, and Christoph Schwarzweller. 2016. On algebraic hierarchies in mathematical repository of Mizar. In *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016*. 363–371. https://doi.org/10.15439/2016F520

[13] Thomas C. Hales, Mark Adams, Gertrud Bauer, Dat Tat Dang, John Harrison, Truong Le Hoang, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Thang Tat Nguyen, Truong Quang Nguyen, Tobias Nipkow, Steven Obua, Joseph Pleso, Jason M. Rute, Alexey Solovyev, An Hoai Thi Ta, Trung Nam Tran, Diep Thi Trieu, Josef Urban, Ky Khac Vu, and Roland Zumkeller. 2015. A formal proof of the Kepler conjecture. *CoRR* abs/1501.02155 (2015). arXiv:1501.02155 http://arxiv.org/abs/1501.02155

[14] Michiel Hazewinkel. 2009. Witt vectors. Part 1. *Handbook of Algebra* (2009), 319–472. https://doi.org/10.1016/s1570-7954(08)00207-6

[15] Artur Korniłowicz and Christoph Schwarzweller. 2014. The First Isomorphism Theorem and Other Properties of Rings. *Formalized Mathematics* 22, **4** (2014), 291–301. https://doi.org/10.2478/forma-2014-0029

[16] Christer Lech. 1953. A note on recurring series. *Ark. Mat.* 2, 5 (08 1953), 417–421. https://doi.org/10.1007/BF02590997

[17] Robert Y. Lewis. 2019. A formal proof of Hensel's lemma over the p-adic integers. In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*. 15–26. https://doi.org/10.1145/3293880.3294089

[18] Robert Y. Lewis and Paul-Nicolas Madelaine. 2020. Simplifying Casts and Coercions. arXiv:2001.10594 [cs.PL]

[19] Assia Mahboubi and Enrico Tassi. 2013. Canonical Structures for the Working Coq User. In *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 7998)*, Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie (Eds.). Springer, 19–34. https://doi.org/10.1007/978-3-642-39634-2_5

[20] Assia Mahboubi and Enrico Tassi. 2017. *Mathematical Components*. https://math-comp.github.io/mcb/.

[21] The mathlib Community. 2020. The Lean Mathematical Library. In *CPP* (New Orleans, LA, USA). ACM, New York, NY, USA, 367–381. https://doi.org/10.1145/3372885.3373824

[22] William McCallum and Bjorn Poonen. 2012. The method of Chabauty and Coleman. In *Explicit methods in number theory*. Panor. Synthèses, Vol. 36. Soc. Math. France, Paris, 99–117.

[23] Álvaro Pelayo, Vladimir Voevodsky, and Michael A. Warren. 2015. A univalent formalization of the p-adic numbers. *Mathematical Structures in Computer Science* 25, 5 (2015), 1147–1171. https://doi.org/10.1017/S0960129514000541

[24] Kazuhiko Sakaguchi. 2020. Validating Mathematical Structures. arXiv. arXiv:2002.00620 [cs.PL] https://arxiv.org/abs/2002.00620

[25] Hermann Ludwig Schmid. 1936. Zyklische algebraische Funktionenkörper vom Grade p^n über endlichem Konstantenkörper der Charakteristik p. *Journal für die reine und angewandte Mathematik* 1936, 175 (1936), 108 – 123. https://doi.org/10.1515/crll.1936.175.108

[26] Daniel Selsam, Sebastian Ullrich, and Leonardo de Moura. 2020. Tabled Typeclass Resolution. arXiv:2001.04301 [cs.PL] https://arxiv.org/abs/2001.04301

[27] Bas Spitters and Eelis van der Weegen. 2011. Type classes for mathematics in type theory. *Mathematical Structures in Computer Science* 21, 4 (2011), 795–825. https://doi.org/10.1017/S0960129511000119

[28] Neil Strickland and Nicola Bellumat. 2019. Iterated chromatic localisation. arXiv:1907.07801 [math.AT]

[29] Philip Wadler and Stephen Blott. 1989. How to Make ad-hoc Polymorphism Less ad-hoc. In *Proceedings of POPL 1989*. 60–76. https://doi.org/10.1145/75277.75283

[30] Yasushige Watase. 2020. Rings of Fractions and Localization. *Formalized Mathematics* 28, **1** (2020), 79–87. https://doi.org/10.2478/forma-2020-0006

[31] Freek Wiedijk. 2007. The QED Manifesto Revisited.

[32] E. Witt. 1937. Zyklische Körper und Algebren der Charakteristik p vom Grad p^n. Struktur diskret bewerteter perfekter Körper mit vollkommenem Restklassenkörper der Charakteristik p. *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1937 (1937), 126 – 140.