Monogenic Extensions of Local Rings

A Formalization Blueprint

Bianca Viray, Bryan Boehnke, Grant Yang, George Peykanu, Tianshuo Wang
UW Math AI Winter 2026

March 18, 2026

Chapter 1

Introduction

This blueprint describes the formalization of Lemmas 3.1 and 3.2 from [1], which concern *monogenic extensions* of local rings. The main results are:

- **Lemma 3.2** (the étale case): A finite injective étale map of local rings is monogenic, and the minimal polynomial has unit derivative.
- **Lemma 3.1** (the partially étale case): If a finite map of local rings admits an étale quotient through a height-one prime, the extension is monogenic.
- **Converse**: A monogenic extension with unit derivative is étale (i.e., standard étale).

The formalization builds on Mathlib's theory of étale ring homomorphisms, local rings, residue fields, and the primitive element theorem.

Bibliography

[1] B. Viray et al., *Monogenic extensions of local rings*, arXiv:2503.07846v2, 2025.

Chapter 2

The Quotient Isomorphism

This chapter establishes that when $R[\beta] = S$ for a finite free extension, the natural map $R[X]/(\operatorname{minpoly}_R \beta) \to S$ is an isomorphism, *without* assuming R is integrally closed.

2.1 Degree Bound via Cayley–Hamilton

Lemma 1 (Minimal polynomial degree bound)**.** *Let $R \to S$ be a finite free extension with R nontrivial. For any $\alpha \in S$,*

$$\deg(\operatorname{minpoly}_R \alpha) \le \operatorname{finrank}_R S.$$

Proof. Let $\ell_\alpha = \operatorname{Algebra.lmul}_R \alpha$ be the left-multiplication endomorphism. By Cayley–Hamilton (`LinearMap.aeval_self_charpoly`), $\operatorname{aeval}_\alpha(\operatorname{charpoly}(\ell_\alpha)) = 0$. Since $\operatorname{charpoly}(\ell_\alpha)$ is monic, the minimal polynomial divides it, so $\deg(\operatorname{minpoly}_R \alpha) \le \deg(\operatorname{charpoly}(\ell_\alpha)) = \operatorname{finrank}_R S$. □

2.2 The Isomorphism via Orzech's Property

Theorem 2 (Quotient isomorphism without integrally closed hypothesis)**.** *Let $R \to S$ be a finite free extension with R nontrivial, and let $\alpha \in S$ with $R[\alpha] = S$. Then $f = \operatorname{minpoly}_R \alpha$ is monic, and the natural map*

$$\varphi \colon R[X]/(f) \longrightarrow S, \qquad X \mapsto \alpha$$

is an R-algebra isomorphism. In particular, S admits an `IsAdjoinRootMonic` *structure for f.*

Proof. The map φ is well-defined since $f(\alpha) = 0$, and surjective since $R[\alpha] = S$.

Rank equality. The upper bound $\deg f \le \operatorname{finrank}_R S$ follows from Lemma 1. For the lower bound, consider the surjection φ. By the quotient rank formula (`finrank_quotient_span_eq_natDegree'`), $\operatorname{finrank}_R(R[X]/(f)) = \deg f$. Since φ is surjective, $\operatorname{finrank}_R S \le \operatorname{finrank}_R(R[X]/(f)) = \deg f$. Hence $\deg f = \operatorname{finrank}_R S$.

Injectivity. Since both $R[X]/(f)$ and S are free R-modules of the same finite rank, and φ is surjective, the composite $e^{-1} \circ \varphi$ is a surjective endomorphism of a finitely generated free module (where e is a linear equivalence from the rank equality). By `OrzechProperty.injective_of_surjective_endomorphism` (which holds for all commutative rings), this composite is injective, hence φ is injective. □

Chapter 3

The Étale Case (Lemma 3.2)

We now prove the main result for finite étale extensions of local rings: such extensions are monogenic, and the minimal polynomial has unit derivative.

3.1 Generator Descends to Residue Field

Lemma 3 (Generator descends to residue fields)**.** *If β generates S over R (i.e., $R[\beta] = S$), then $\bar{\beta} = \beta \bmod \mathfrak{m}_S$ generates k_S over k_R.*

Proof. Given $x \in k_S$, lift to $s \in S$, express $s = p(\beta)$ for some $p \in R[X]$, then $x = \bar{p}(\bar{\beta})$ where $\bar{p} = p \bmod \mathfrak{m}_R \in k_R[X]$. The key commutativity ($\mathrm{residue}_S \circ \varphi = \mathrm{algebraMap}_{k_R \to k_S} \circ \mathrm{residue}_R$) is used via `map_aeval_eq_aeval_map`. ☐

3.2 Rank Equality across Residue Fields

Lemma 4 (Rank preserved by residue field base change)**.** *If $R \to S$ is a finite étale extension of local rings with faithful scalar action, then*

$$\mathrm{finrank}_R\, S = \mathrm{finrank}_{k_R}\, k_S.$$

Proof. Since $R \to S$ is étale, it is smooth, hence flat. Finite + flat over a local ring implies free (`Module.free_of_flat_of_isLocalRing`).

The étale condition gives $\mathfrak{m}_R \cdot S = \mathfrak{m}_S$ (`Algebra.FormallyUnramified.map_maximalIdeal`), so $S/\mathfrak{m}_R S \cong k_S$. By `IsLocalRing.finrank_quotient_map`, $\mathrm{finrank}_{k_R}(S/\mathfrak{m}_R S) = \mathrm{finrank}_R\, S$. The quotient equivalence from $\mathfrak{m}_R S = \mathfrak{m}_S$ transfers the finrank. ☐

3.3 Minimal Polynomial Descends to Residue Field

Lemma 5 (Minimal polynomial maps to residue minimal polynomial)**.** *Let $R \to S$ be a finite étale extension of local rings with faithful scalar action, and $\beta \in S$ with $R[\beta] = S$. Then*

$$(\mathrm{minpoly}_R\, \beta) \bmod \mathfrak{m}_R = \mathrm{minpoly}_{k_R}\, \bar{\beta}.$$

Proof. Set $f = \mathrm{minpoly}_R\, \beta$, $\bar{f} = f \bmod \mathfrak{m}_R$, $g = \mathrm{minpoly}_{k_R}\, \bar{\beta}$.

Since $\bar{\beta}$ is a root of $\bar{f}$ (by `hom_eval`$_2$), we have $g \mid \bar{f}$. Both f and g are monic, and $\bar{f}$ is monic. The degree chain:

$$\begin{aligned}
\deg \bar{f} &= \deg f && \text{(monic polynomial preserves degree under reduction)}\\
&= \operatorname{finrank}_R S && \text{(Theorem 2)}\\
&= \operatorname{finrank}_{k_R} k_S && \text{(Lemma 4)}\\
&= [k_R(\bar{\beta}) : k_R] && \text{(Lemma 3: } k_R[\bar{\beta}] = k_S\text{)}\\
&= \deg g && \text{(degree of minimal polynomial = extension degree)}
\end{aligned}$$

Since $g \mid \bar{f}$, both are monic, and they have the same degree, $\bar{f} = g$. ☐

3.4 Unit Derivative Condition

Theorem 6 (Derivative of minimal polynomial is a unit)**.** *Let $R \to S$ be a finite étale extension of local rings with faithful scalar action, and $\beta \in S$ with $R[\beta] = S$. Then*

$$f'(\beta) \in S^\times$$

where $f = \operatorname{minpoly}_R \beta$.

Proof. Since S is a local ring, it suffices to show $\overline{f'(\beta)} \neq 0$ in k_S (`IsLocalRing.residue_ne_zero_iff_isUnit`).

Since $R \to S$ is étale, the residue field extension $k_R \to k_S$ is separable. By Lemma 5, $\bar{f} = \operatorname{minpoly}_{k_R} \bar{\beta}$, so $\bar{f}$ is separable (`Algebra.IsSeparable.isSeparable`). This means $\bar{f}'(\bar{\beta}) \neq 0$ (`Separable.aeval_derivative_ne_zero`).

By `hom_eval`$_2$, $\overline{f'(\beta)} = \bar{f}'(\bar{\beta}) \neq 0$. ☐

3.5 Existence of Generator (Nakayama Argument)

Theorem 7 (Finite étale local extensions are monogenic)**.** *Let $R \to S$ be a finite étale extension of local rings with faithful scalar action. Then there exists $\beta \in S$ such that $R[\beta] = S$.*

Proof. By the primitive element theorem (`Field.exists_primitive_element`), there exists $\beta_0 \in k_S$ with $k_R[\beta_0] = k_S$. Lift β_0 to $\beta \in S$ via `Ideal.Quotient.mk_surjective`.

It suffices to show $\operatorname{Algebra.adjoin} R\{\beta\} = \top$. By Nakayama's lemma (`Submodule.le_of_le_smul_of_le_jacobson_bot`), it suffices to show $S \subseteq R[\beta] + \mathfrak{m}_R \cdot S$.

For any $s \in S$, the residue $\bar{s} \in k_S = k_R[\beta_0]$, so there exists $t_0 \in R[\beta]$ with $\overline{t_0} = \bar{s}$. Then $s - t_0 \in \mathfrak{m}_S = \mathfrak{m}_R \cdot S$ (using the étale condition `Algebra.FormallyUnramified.map_maximalIdeal`). ☐

Chapter 4

Converse: Monogenic with Unit Derivative Implies Étale

Theorem 8 (Standard étale from monogenic with unit derivative)**.** *Let* $f \in R[X]$ *be monic with* $S \cong R[X]/(f)$ *(via an* `IsAdjoinRootMonic` *structure). If* $f'(\beta) \in S^\times$ *where* β *is the image of* X*, then* $R \to S$ *is étale.*

Proof. Construct a `StandardEtalePair` using (f, f'): the Bézout relation $f' \cdot 1 + f \cdot 0 = (f')^1$ is trivially satisfied.

The natural map $P.\texttt{lift}\colon P.\texttt{Ring} \to S$ sending $X \mapsto \beta$ is well-defined by $f(\beta) = 0$ and the unit condition on $f'(\beta)$.

Construct an explicit inverse via `AdjoinRoot.liftAlgHom` composed with the algebra equivalence $S \cong$ `AdjoinRoot` f. Verify it is both a left and right inverse by checking on generators, concluding bijectivity.

Hence S is standard étale (`Algebra.IsStandardEtale`), and étale follows. □

Chapter 5

The Partially Étale Case (Lemma 3.1)

This chapter proves the more general Lemma 3.1: monogenicity from an étale quotient through a height-one prime. The hypotheses are stronger on the rings (domains, integrally closed, UFD) but weaker on the map (only a quotient needs to be étale).

5.1 Sub-lemmas

Lemma 9 (Height-one primes are principal in UFDs)**.** *In a UFD S, every prime ideal $\mathfrak{q}$ of height 1 is principal.*

Proof. Since $\mathfrak{q} \neq 0$, it contains an irreducible element p. Then $\operatorname{span}\{p\}$ is prime and contained in $\mathfrak{q}$. If $\operatorname{span}\{p\} \subsetneq \mathfrak{q}$, then $\operatorname{height}(\operatorname{span}\{p\}) < \operatorname{height}(\mathfrak{q}) = 1$, forcing $\operatorname{span}\{p\} = 0$, contradicting $p \neq 0$. Hence $\mathfrak{q} = \operatorname{span}\{p\}$. ☐

Lemma 10 (Taylor expansion for polynomial evaluation)**.** *For any polynomial $f \in R[X]$ and elements $x, h \in S$, there exists $c \in S$ such that*

$$f(x+h) = f(x) + f'(x) \cdot h + h^2 \cdot c.$$

Proof. Reduce modulo $\operatorname{span}\{h^2\}$ using `Polynomial.aeval_add_of_sq_eq_zero`, then lift. ☐

Lemma 11 (Maximal ideal decomposition from étale quotient)**.** *Let R and S be local rings with S a finite R-algebra, and let $\mathfrak{q} \subset S$ be a prime ideal. If the quotient map $R/(\mathfrak{q} \cap R) \to S/\mathfrak{q}$ is étale, then*

$$\mathfrak{m}_S = \mathfrak{q} + \mathfrak{m}_R \cdot S.$$

Proof. The quotient rings are local domains, and the étale condition on $R/\mathfrak{p} \to S/\mathfrak{q}$ gives $\mathfrak{m}_{R/\mathfrak{p}} \cdot (S/\mathfrak{q}) = \mathfrak{m}_{S/\mathfrak{q}}$ by `Algebra.FormallyUnramified.map_maximalIdeal`. Pulling back through the quotient maps, using $\operatorname{map}(\operatorname{mk}\ \mathfrak{q})(\mathfrak{m}_S) = \mathfrak{m}_{S/\mathfrak{q}}$ and the analogous statement for R, the equality lifts to $\mathfrak{m}_S = \mathfrak{q} + \mathfrak{m}_R S$. ☐

Lemma 12 (Quotient adjustment)**.** *Let $\mathfrak{q} \subset S$ be an ideal and $\mathfrak{q} = \operatorname{span}\{q_0\}$. Then for any $\beta \in S$, the image of $\beta + q_0$ in $S/\mathfrak{q}$ is equal to the image of β.*

Proof. Since $(\beta + q_0) - \beta = q_0 \in \mathfrak{q}$, the elements are congruent modulo $\mathfrak{q}$. ☐

5.2 Nakayama Helpers

Lemma 13 (Quotient lifting)**.** *Let* $\mathfrak{q} \subset S$ *be an ideal and* $\beta \in S$*. If* β *generates* $S/\mathfrak{q}$ *over* $R/(\mathfrak{q} \cap R)$*, then for every* $s \in S$ *there exists* $t \in R[\beta]$ *with* $s - t \in \mathfrak{q}$*.*

Proof. Since $(R/\mathfrak{p})[\bar{\beta}] = S/\mathfrak{q}$, the map $\mathrm{aeval}_{\bar{\beta}}$ is surjective. Lift a polynomial preimage of $\bar{s}$ through `Polynomial.map_surjective` to obtain $t \in R[\beta]$ with $s - t \in \mathfrak{q}$. ☐

Lemma 14 (Nakayama argument for quotient generation)**.** *Let* R, S *be local rings with* S *a finite* R*-algebra, and* $\mathfrak{q} \subset S$ *a prime ideal. Let* $\beta \in S$ *and* $\pi \in R[\beta]$*. Assume:*

1. $\bar{\beta}$ *generates* $S/\mathfrak{q}$ *over* $R/(\mathfrak{q} \cap R)$*,*
2. $\mathfrak{m}_S = \mathrm{span}\{\pi\} + \mathfrak{m}_R S$*.*

Then $R[\beta] = S$*.*

Proof. Set $A = R[\beta]$. The quotient $S/\mathfrak{m}_R S$ is Artinian (finite over the field k_R), so $\mathfrak{m}_S^n \subseteq \mathfrak{m}_R S$ for some n.

Power bound. By induction, $\mathfrak{m}_S^k \subseteq \mathrm{span}\{\pi^k\} + \mathfrak{m}_R S$, using the hypothesis on $\mathfrak{m}_S$.

Iterative reduction. For $x \in \mathfrak{q}$, by repeated application of quotient lifting (Lemma 13) and the power bound, we find $a \in A$ with $x - a \in \mathrm{span}\{\pi^n\} + \mathfrak{m}_R S \subseteq \mathfrak{m}_R S$.

Hence $\mathfrak{q} \subseteq A + \mathfrak{m}_R S$ as R-submodules. Combined with the quotient lifting for general elements, $S \subseteq A + \mathfrak{m}_R S$. By Nakayama's lemma, $A = S$. ☐

5.3 Principal Adjustment Lemma

Lemma 15 (Adjustment by principal generator)**.** *Let* $(R, \mathfrak{m}_R)$ *and* $(S, \mathfrak{m}_S)$ *be local domains with* R *integrally closed and* S *a finite* R*-algebra with faithful scalar action. Let* $\mathfrak{q} \subset S$ *be a prime ideal with* $\mathfrak{q} = (q_0)$*, and let* $\beta \in S$*,* $f_1 \in R[X]$*,* $a \in S$ *satisfy:*

1. $f_1(\beta) = q_0 \cdot a$ *with* $a \in \mathfrak{m}_S$*,*
2. $f_1'(\beta) \notin \mathfrak{m}_S$*,*
3. $\mathfrak{m}_S = \mathfrak{q} + \mathfrak{m}_R S$*,*
4. $\bar{\beta}$ *generates* $S/\mathfrak{q}$ *over* $R/(\mathfrak{q} \cap R)$*.*

Then there exist a monic $f \in R[X]$ *and an* `IsAdjoinRootMonic` *structure for* S *over* f*.*

Proof. Set $\beta' = \beta + q_0$. By Lemma 10, $f_1(\beta') = q_0 \cdot (a + f_1'(\beta) + q_0 b)$ for some $b \in S$. Since $f_1'(\beta) \notin \mathfrak{m}_S$ and $a, q_0 b \in \mathfrak{m}_S$ (the former by hypothesis), the cofactor is a unit. Hence $\mathrm{span}\{f_1(\beta')\} = \mathfrak{q}$, and $\mathfrak{m}_S = \mathrm{span}\{f_1(\beta')\} + \mathfrak{m}_R S$.

Since $\beta' \equiv \beta \pmod{\mathfrak{q}}$ (Lemma 12), it still generates the quotient, so Lemma 14 gives $R[\beta'] = S$. Finally, Theorem 2 provides the `IsAdjoinRootMonic` structure for $\mathrm{minpoly}_R\, \beta'$. ☐

5.4 Main Theorem: Lemma 3.1

Theorem 16 (Monogenicity from étale height-one quotient)**.** *Let* $(R, \mathfrak{m}_R)$ *and* $(S, \mathfrak{m}_S)$ *be local domains with* R *integrally closed and* S *a UFD. Let* $R \to S$ *be a finite injective ring homomorphism. If there exists a prime ideal* $\mathfrak{q} \subset S$ *of height 1 such that the induced map* $R/(\mathfrak{q} \cap R) \to S/\mathfrak{q}$ *is étale, then there exist a monic* $f \in R[X]$ *and an* `IsAdjoinRootMonic` *structure for* S *over* f*.*

Proof. **Case 1: $R \to S$ is already étale.** Apply Theorem 7 to find β with $R[\beta] = S$, then Theorem 2 for the isomorphism.

Case 2: $R \to S$ is not étale. Apply Theorem 7 to the étale quotient $R_0 = R/\mathfrak{p} \to S_0 = S/\mathfrak{q}$ to get $\beta_0 \in S_0$ with $R_0[\beta_0] = S_0$. Lift to $\beta \in S$.

By Lemma 9, $\mathfrak{q} = (q_0)$ for some $q_0 \in S$. Lift the minimal polynomial $f_0 = \operatorname{minpoly}_{R_0} \beta_0$ to a monic $f_1 \in R[X]$. By Lemma 11, $\mathfrak{m}_S = \mathfrak{q} + \mathfrak{m}_R S$.

Sub-case 2a: If $f_1(\beta)$ generates $\mathfrak{m}_S$ modulo $\mathfrak{m}_R S$, apply Lemma 14 directly to conclude $R[\beta] = S$.

Sub-case 2b: Otherwise, since $f_1(\beta) \in \mathfrak{q} = (q_0)$, write $f_1(\beta) = q_0 \cdot a$. If a were a unit, then $\operatorname{span}\{f_1(\beta)\} = \mathfrak{q}$ and Sub-case 2a would apply; so $a \in \mathfrak{m}_S$. The derivative condition $f_1'(\beta) \notin \mathfrak{m}_S$ follows from Theorem 6 applied to the étale quotient. Now apply Lemma 15 with q_0, β, f_1, and a. □

Chapter 6

Plan for Mathlib PRs

This chapter outlines a strategy for upstreaming the formalized results into Mathlib via a sequence of small, self-contained pull requests. Each PR should be independently reviewable and useful.

Throughout, we reference Mathlib file paths as of the current toolchain. The key Mathlib files involved are:

- `Mathlib/RingTheory/IsAdjoinRoot.lean` — defines `IsAdjoinRoot`, `IsAdjoinRootMonic`, and `finrank_quotient_span_eq_natDegree'`. Imports `FieldTheory.Minpoly.IsIntegrallyClosed` and `RingTheory.PowerBasis`. Does *not* transitively import `LinearAlgebra.Charpoly.Basic`.
- `Mathlib/RingTheory/Etale/Basic.lean` — defines `Algebra.Etale`.
- `Mathlib/RingTheory/Etale/StandardEtale.lean` — defines `StandardEtalePair`, `StandardEtalePresentation`, and `Algebra.IsStandardEtale`.
- `Mathlib/RingTheory/Unramified/LocalRing.lean` — contains `FormallyUnramified.map_maximalIdeal`.
- `Mathlib/RingTheory/LocalRing/ResidueField/` — `Defs.lean` (defines `residue`, `ResidueField`), `Basic.lean` (surjectivity, kernel), `Instances.lean` (separability instances).
- `Mathlib/RingTheory/LocalRing/Quotient.lean` — contains `finrank_quotient_map` and `exists_maximalIdeal_pow_le_of_isArtinianRing_quotient`.
- `Mathlib/RingTheory/LocalRing/Module.lean` — contains `Module.free_of_flat_of_isLocalRing`.
- `Mathlib/RingTheory/Ideal/Height.lean` — defines `Ideal.height`.
- `Mathlib/FieldTheory/PrimitiveElement.lean` — contains `Field.exists_primitive_element`.
- `Mathlib/RingTheory/Nakayama.lean` — contains `Submodule.le_of_le_smul_of_le_jacobson_bot`.
- `Mathlib/Algebra/Polynomial/Taylor.lean` — contains `aeval_add_of_sq_eq_zero`.

6.1 PR 1: Quotient Isomorphism without Integrally Closed

Declarations

- `minpoly.natDegree_le'`: For a finite free extension with R nontrivial, $\deg(\mathrm{minpoly}_R\,\alpha) \leq \mathrm{finrank}_R\,S$. Generalizes the existing `minpoly.natDegree_le` in `FieldTheory/IntermediateField/Adjoin/Basic.lean` (which requires K, L to be fields with `FiniteDimensional`).
- `IsAdjoinRootMonic.mkOfAdjoinEqTop'`: If $R[\alpha] = S$ and S is finite free over R, then S admits an `IsAdjoinRootMonic` structure for $\mathrm{minpoly}_R\,\alpha$.

File placement

Option A (preferred): Both declarations go into `Mathlib/RingTheory/IsAdjoinRoot.lean`. This is where `IsAdjoinRootMonic`, `finrank_quotient_span_eq_natDegree'`, and the existing adjoin-root machinery live.

Import issue: `natDegree_le'` uses `LinearMap.aeval_self_charpoly` and `charpoly_natDegree` from `Mathlib.LinearAlgebra.Charpoly.Basic`, which is *not* a transitive import of `IsAdjoinRoot.lean`. Adding this import may be acceptable since the file already has heavy imports (`PowerBasis`, `Minpoly.IsIntegrallyClosed`), but reviewers may prefer a separate file.

Option B: Put `natDegree_le'` in a new file `Mathlib/FieldTheory/Minpoly/Basic.lean` (or `Mathlib/RingTheory/Polynomial/Minpoly/Degree.lean`), then import it from `IsAdjoinRoot.lean` for `mkOfAdjoinEqTop'`.

Namespace and naming

- `minpoly.natDegree_le'`: Currently uses `_root_` prefix to escape the `Monogenic` namespace. In Mathlib, consider renaming to `minpoly.natDegree_le_finrank` or similar, and whether to replace the existing field-level `minpoly.natDegree_le` or keep both.
- `IsAdjoinRootMonic.mkOfAdjoinEqTop'`: Already in the correct namespace (matching the Mathlib original `IsAdjoinRootMonic.mkOfAdjoinEqTop`). For Mathlib, the prime suffix should be dropped; the existing Mathlib version requires `IsIntegrallyClosed`, so this would replace it.

Required imports (for the declarations)

- `Mathlib.LinearAlgebra.Charpoly.Basic` (for `aeval_self_charpoly`, `charpoly_natDegree`, `charpoly_monic`).
- `Mathlib.RingTheory.OrzechProperty` (for `injective_of_surjective_endomorphism`; likely already transitive via `IsAdjoinRoot`).
- `Mathlib.RingTheory.IsAdjoinRoot` (for `IsAdjoinRootMonic`, `AdjoinRoot.liftAlgHom`, `finrank_quotient_span_eq_natDegree'`).

Proof adjustments for Mathlib style

- The current proof of `natDegree_le'` uses `classical` and local `let` bindings for basis and matrix. This should be rewritten to avoid `classical` if possible (use `Decidable` instances or restructure).
- Drop the `Monogenic` namespace wrapper.
- Add docstrings in Mathlib format.
- The current proof uses `Module.finrank_eq_card_chooseBasisIndex`; check this is still the canonical spelling.

Potential issues

- Import weight: adding `Charpoly.Basic` to the import tree of `IsAdjoinRoot.lean` may increase compile times. Measure before submitting.
- The existing `minpoly.natDegree_le` for fields is proved via `IntermediateField.adjoin.finrank`. Reviewers may ask if the field version should be deduced from the ring version or kept separate.

6.2 PR 2: Residue Field Properties for Étale Local Extensions

Declarations

- `adjoin_residue_eq_top_of_adjoin_eq_top`: If $R[\beta] = S$ then $k_R[\bar{\beta}] = k_S$.
- `finrank_eq_finrank_residueField`: $\operatorname{finrank}_R S = \operatorname{finrank}_{k_R} k_S$ for finite étale local extensions.
- `minpoly_map_residue`: For a finite étale local extension with $R[\beta] = S$: $(\operatorname{minpoly}_R \beta) \bmod \mathfrak{m}_R = \operatorname{minpoly}_{k_R} \bar{\beta}$.
- `isUnit_aeval_derivative_minpoly_of_adjoin_eq_top`: For a finite étale local extension with $R[\beta] = S$, $f'(\beta) \in S^\times$ where $f = \operatorname{minpoly}_R \beta$.

File placement

Preferred: A new file `Mathlib/RingTheory/LocalRing/ResidueField/Minpoly.lean` or `Mathlib/RingTheory/Etale/Monogenic.lean` that collects these cohesive residue field properties for étale local extensions.

Namespace

`IsLocalRing` or `Algebra.Etale` namespace.

Required imports

- `Mathlib.RingTheory.LocalRing.ResidueField.Defs` (for `residue`, `ResidueField`).
- `Mathlib.RingTheory.Unramified.LocalRing` (for `FormallyUnramified.map_maximalIdeal`, used by `finrank_eq_finrank_residueField`).
- `Mathlib.RingTheory.Smooth.Flat` (for `Algebra.Smooth.flat`, used in the étale ⇒ free chain).
- `Mathlib.RingTheory.LocalRing.Module` (for `free_of_flat_of_isLocalRing`).
- `Mathlib.RingTheory.LocalRing.Quotient` (for `finrank_quotient_map`).
- `IsAdjoinRootMonic.mkOfAdjoinEqTop'` from PR 1 (used in the degree chain: $\deg f = \operatorname{finrank}_R S$ via `.finrank`).
- `Mathlib.FieldTheory.IntermediateField.Adjoin.Basic` (for `IntermediateField.adjoin.finrank`, `adjoin_simple_toSubalgebra_of_isAlgebraic`).
- `Mathlib.FieldTheory.Separable` (for `Algebra.IsSeparable.isSeparable`, `Separable.aeval_derivative_ne_zero`).

Proof structure

- **Residue generator & rank:** `finrank_eq_finrank_residueField` uses étale ⇒ smooth ⇒ flat ⇒ free.
- **Minimal polynomial descent:** `minpoly_map_residue` compares degrees: $g \mid \bar{f}$, both monic, same degree ⇒ equal. The degree chain uses `mkOfAdjoinEqTop'.finrank` (PR 1), `finrank_eq_finrank_residueField`, and `IntermediateField.adjoin.finrank` (Mathlib).
- **Unit derivative:** Reduce to residue field via `minpoly_map_residue`, apply separability of $k_R \to k_S$ (from étale), conclude $\bar{f}'(\bar{\beta}) \neq 0$, lift back to unit via `IsLocalRing.residue_ne_zero_iff_isUnit`.

Potential issues

- `finrank_eq_finrank_residueField` uses the étale ⇒ smooth ⇒ flat chain. Consider whether to state it more generally for "finite free local extensions where $\mathfrak{m}_R S = \mathfrak{m}_S$".
- The étale hypothesis is used only to get $\operatorname{finrank}_R S = \operatorname{finrank}_{k_R} k_S$. If this equality holds more generally, the `minpoly_map_residue` lemma could be stated without `Algebra.Etale`. Reviewers may ask for this generalization.
- The hypothesis chain étale ⇒ separable residue field should be automatic, but verify the instance path works: `Algebra.Etale R S` ⇒ `Algebra.FormallyUnramified R S` ⇒ `Algebra.IsSeparable kR kS`.
- Uses `eq_of_monic_of_dvd_of_natDegree_le` — verify this is still the canonical Mathlib name.

6.3 PR 3: Finite Étale Local Extensions are Monogenic

Declarations

- `exists_adjoin_eq_top`: For a finite étale local extension $R \to S$ with faithful scalar action, $\exists \beta \in S,\ R[\beta] = S$.

File placement

Preferred: New file `Mathlib/RingTheory/Etale/Monogenic.lean` or `Mathlib/RingTheory/Etale/LocalRing.lean`. This is a natural home for étale-specific results about local rings.

Namespace

`Algebra.Etale.exists_adjoin_eq_top` or just in the root namespace with appropriate variable scoping.

Required imports

- `Mathlib.FieldTheory.PrimitiveElement` (for `Field.exists_primitive_element`).
- `Mathlib.RingTheory.Unramified.LocalRing` (for `FormallyUnramified.map_maximalIdeal`).
- `Mathlib.RingTheory.Nakayama` (for `Submodule.le_of_le_smul_of_le_jacobson_bot`).
- `Mathlib.RingTheory.LocalRing.ResidueField.Defs` (for `residue`, `ResidueField`).

Proof structure

Primitive element theorem gives $\beta_0 \in k_S$ with $k_R[\beta_0] = k_S$. Lift to $\beta \in S$. Show $S \subseteq R[\beta] + \mathfrak{m}_R S$: for each $s \in S$, lift a polynomial preimage of $\bar{s}$ through `Polynomial.map_surjective` to find $t \in R[\beta]$ with $s - t \in \mathfrak{m}_S = \mathfrak{m}_R S$. Conclude by Nakayama.

Potential issues

- This PR is *independent* of PRs 1 and 2. It does not need `mkOfAdjoinEqTop'` or `minpoly_map_residue`; it only produces a generator β, not an isomorphism $S \cong R[X]/(f)$.
- The proof uses `IntermediateField.adjoin_simple_toSubalgebra_of_isAlgebraic` to convert between `Algebra.adjoin` and `IntermediateField.adjoin` in the residue field. This is a common pattern but sometimes fragile with universe issues.
- The `FaithfulSMul R S` hypothesis is equivalent to injectivity of φ. Reviewers may prefer `IsLocalHom (algebraMap R S)` or just `Function.Injective (algebraMap R S)`. Note `faithfulSMul_iff_algebraMap_injective` exists in Mathlib.

6.4 PR 4: Converse — Standard Étale from Unit Derivative

Declarations

- `IsAdjoinRootMonic.algebra_etale`: If $S \cong R[X]/(f)$ (via `IsAdjoinRootMonic`) and $f'(\beta) \in S^\times$, then $R \to S$ is étale.

File placement

Preferred: `Mathlib/RingTheory/Etale/StandardEtale.lean`, directly after the definition of `StandardEtalePair` and `Algebra.IsStandardEtale`. This is where the standard étale machinery lives and the proof constructs a `StandardEtalePair`.

Namespace

`IsAdjoinRootMonic.algebra_etale` (already correctly namespaced).

Required imports

- `Mathlib.RingTheory.Etale.StandardEtale` (for `StandardEtalePair`, `HasMap`, `lift`, `hom_ext`, `Algebra.IsStandardEtale`).
- `Mathlib.RingTheory.IsAdjoinRoot` (for `IsAdjoinRootMonic`, `adjoinRootAlgEquiv`, `map_surjective`).

Both of these should already be in the import tree of `StandardEtale.lean`, so no new imports are needed if placed there.

Proof structure

Construct `StandardEtalePair R` with $f, f', 1, 0, 1$ and the trivial Bézout relation $f' \cdot 1 + f \cdot 0 = (f')^1$. Build an explicit inverse of `P.lift` by composing `AdjoinRoot.liftAlgHom` with `adjoinRootAlgEquiv.symm`. Verify left/right inverse on generators via `P.hom_ext`.

Potential issues

- This theorem essentially says "standard étale presentations arise from adjoin-root-monic with unit derivative." Mathlib may already have something close or the reviewers may want to restructure. Check for overlap with existing material in `StandardEtale.lean`.
- The proof uses `AdjoinRoot.liftAlgHom` which was recently introduced as a replacement for the deprecated `AdjoinRoot.liftHom`. Verify this is still the current API.

6.5 PR 5: Nakayama Helpers and Partially Étale Case (Lemma 3.1)

This is the most complex PR. It should likely be split into several sub-PRs.

Sub-PR 5a: Height-One Primes in UFDs are Principal

Declaration: `Ideal.exists_span_singleton_eq_of_prime_of_height_one`

File: `Mathlib/RingTheory/Ideal/Height.lean` (append to existing file).

Imports: Only needs `Ideal.height`, `IsPrime`, `UniqueFactorizationMonoid`. All already in that file's import tree.

Notes: Self-contained, no dependencies on other PRs. This is a basic fact about UFDs that should be independently useful. The proof uses `IsPrime.exists_mem_prime_of_ne_bot`, `height_strict_mono_of_is_prime`, and `primeHeight_eq_zero_iff`.

Sub-PR 5b: Taylor Expansion for Polynomial Evaluation

Declaration: `exists_aeval_add_eq`: $f(x + h) = f(x) + f'(x)h + h^2c$.

File: `Mathlib/Algebra/Polynomial/Taylor.lean` (append to existing file, which already has `aeval_add_of_sq_eq_zero`).

Imports: Only needs `Polynomial.Taylor` (already there) and `Ideal.Quotient` basics.

Notes: Self-contained. The proof reduces mod $\langle h^2 \rangle$ using `aeval_add_of_sq_eq_zero`, then lifts via `Ideal.mem_span_singleton`. Very clean.

Sub-PR 5c: Maximal Ideal Decomposition and Quotient Lifting

Declarations:

- `maximalIdeal_eq_sup_of_etale_quotient`: If $R/\mathfrak{p} \to S/\mathfrak{q}$ is étale, then $\mathfrak{m}_S = \mathfrak{q} + \mathfrak{m}_R S$.
- `exists_adjoin_sub_mem`: Quotient lifting.
- `adjoin_eq_top_of_quotient`: Nakayama with quotient generation.

File: New file, perhaps `Mathlib/RingTheory/LocalRing/Quotient/Etale.lean` or `Mathlib/RingTheory/Etale/LocalRing.lean`.

Imports:

- `Mathlib.RingTheory.Unramified.LocalRing` (for `FormallyUnramified.map_maximalIdeal`).
- `Mathlib.RingTheory.RingHom.Etale` (for `RingHom.Etale`).
- `Mathlib.RingTheory.LocalRing.Quotient` (for `exists_maximalIdeal_pow_le_of_isArtinianRing_quotient`).
- `Mathlib.RingTheory.Nakayama`.

Notes:

- `maximalIdeal_eq_sup_of_etale_quotient` has a somewhat long proof involving quotient maps and the commutative square map(mk $\mathfrak{q}$). It uses `Ideal.map_eq_iff_sup_ker_eq_of_surjective` which may be fragile.
- `adjoin_eq_top_of_quotient` is the most intricate proof in the project (Artinian descent + iterative reduction + Nakayama). It is already decomposed into helper lemmas: `Ideal.pow_le_span_pow_sup` (pure ideal arithmetic) and `exists_sub_mem_adjoin_of_pow` (iterative approximation).
- `exists_adjoin_sub_mem` lifts polynomials through the quotient map via `Polynomial.map_surjective`.

Sub-PR 5d: Main Theorem (Lemma 3.1)

Declarations:

- `exists_isAdjoinRootMonic_of_principal_adjust`: The principal adjustment sub-case (Case 2b), extracted as a standalone lemma.
- `exists_isAdjoinRootMonic_of_quotientMap_etale`: The main theorem.

File: Same as 5c, or a dedicated file `Mathlib/RingTheory/Etale/Monogenic.lean`.
Dependencies: PRs 1, 2, 3, and sub-PRs 5a–5c.
Imports (beyond 5c):

- `Mathlib.RingTheory.Ideal.Height` (for height-one primes, 5a).
- `Mathlib.Algebra.Polynomial.Taylor` (for Taylor expansion, 5b).
- `Mathlib.Algebra.Polynomial.Lifts` (for `lifts_and_degree_eq_and_monic`, used to lift $f_0 \in R_0[X]$ to monic $f_1 \in R[X]$).
- PR 1 (`mkOfAdjoinEqTop'`, used to build the isomorphism).
- PR 2 (`isUnit_aeval_derivative_minpoly_of_adjoin_eq_top`, for the derivative non-vanishing in Case 2b).
- PR 3 (`exists_adjoin_eq_top`, applied to the étale quotient).

Proof structure: Case split on whether $R \to S$ is already étale. Case 1 uses PR 3 + PR 1 directly. Case 2 lifts a generator from the étale quotient, then either (2a) applies the Nakayama argument directly, or (2b) extracts a with $f_1(\beta) = q_0 \cdot a$ and shows $a \in \mathfrak{m}_S$ (if a were a unit, Sub-case 2a would apply), then delegates to `exists_isAdjoinRootMonic_of_principal_adjust`.
Notes:

- The hypotheses include `IsIntegrallyClosed R` and `UniqueFactorizationMonoid S`. These are strong; reviewers may ask whether they can be weakened.
- An alternative formulation with an explicit ring homomorphism (transferring between Algebra. adjoin R and Algebra. adjoin $\varphi(R)$) could be added as a trivial corollary if needed.

6.6 Dependency Graph Summary

The PR dependency structure is:

$$\begin{tikzcd}
\text{PR 1} \arrow[d] & \text{PR 3} \arrow[d] & \text{PR 4} \\
\text{PR 2} \arrow[r] & \text{PR 5} &
\end{tikzcd}$$

PRs 1, 3, and 4 can be submitted independently and in parallel. PR 2 depends on PR 1. PR 5 depends on PRs 1, 2, and 3. Sub-PRs 5a and 5b are fully independent and can be submitted first.