1. Ring Basic Lemmas

1.1. Overview

Lemmas about semirings, rings and domains, focusing on the interaction between + and *.
Definitions are in `Algebra.Ring.Defs`.

1.2. AddHom Multiplication

1.2.1. Left and Right Multiplication

`AddHom.mulLeft [Distrib R] (r : R)`: Left multiplication by `r` as additive homomorphism
- `toFun := (r * ·)`
- Preserves addition: `r * (a + b) = r * a + r * b`

`AddHom.mulRight [Distrib R] (r : R)`: Right multiplication by `r` as additive homomorphism
- `toFun := (· * r)`
- Preserves addition: `(a + b) * r = a * r + b * r`

1.3. AddMonoidHom Multiplication

1.3.1. Basic Multiplication Homomorphisms

`AddMonoidHom.mulLeft (r : R)`: Left multiplication in semirings
- Maps zero: `r * 0 = 0`
- Maps addition: `r * (a + b) = r * a + r * b`

`AddMonoidHom.mulRight (r : R)`: Right multiplication in semirings
- Maps zero: `0 * r = 0`
- Maps addition: `(a + b) * r = a * r + b * r`

1.3.2. Bilinear Multiplication

`AddMonoidHom.mul : R →+ R →+ R`: Multiplication as additive homomorphism in both arguments
- First argument: `mul x` gives left multiplication by `x`
- Curried form makes multiplication bilinear

1.3.3. Preservation Lemmas

`map_mul_iff`: Characterizes when `f : R →+ S` preserves multiplication
- `(∀ x y, f (x * y) = f x * f y)` iff composition with `mul` commutes

`mulLeft_eq_mulRight_iff_forall_commute`:
- `mulLeft a = mulRight a` iff a commutes with everything

1.4. AddMonoid.End Multiplication

1.4.1. Endomorphism Multiplication

`AddMonoid.End.mulLeft : R →+ AddMonoid.End R`:
- Sends `a` to the endomorphism of left multiplication by `a`

`AddMonoid.End.mulRight : R →+ AddMonoid.End R`:
- Sends `a` to the endomorphism of right multiplication by `a`

1.4.2. Commutativity

`mulRight_eq_mulLeft` [NonUnitalNonAssocCommSemiring R]:
- In commutative semirings, left and right multiplication coincide

1.5. HasDistribNeg

1.5.1. Opposite Negation

`MulOpposite.instHasDistribNeg`: Distributive negation on opposite

- `neg_mul`: Negation distributes over multiplication from left
- `mul_neg`: Negation distributes over multiplication from right

1.6. Special Results

1.6.1. Vieta's Formula

`vieta_formula_quadratic`: For quadratic $x^2 - bx + c = 0$

- If `x` is a root, then exists `y` with:
  - `x + y = b`
  - `x * y = c`

1.7. Implementation Notes

- Uses `@[simps]` for automatic simplification lemma generation
- Coercion lemmas marked with `@[simp, norm_cast]`
- Local simp attributes for specific proofs (e.g., commutativity in Vieta)