1. Basic Group Lemmas

1.1. Overview

Basic lemmas about semigroups, monoids, and groups. Most are one-line proofs from axioms. Definitions are in `Algebra/Group/Defs.lean`.

1.2. Conditional Expressions

1.2.1. Power with if-then-else

`pow_ite`, `ite_pow`: Distribute powers over conditionals

- `a ^ (if P then b else c) = if P then a ^ b else a ^ c`
- `(if P then a else b) ^ c = if P then a ^ c else b ^ c`

`pow_dite`, `dite_pow`: Dependent versions with proof-carrying conditionals

All lemmas have additive versions (`smul_ite`, etc.) via `@[to_additive]`.

1.3. Semigroup Properties

1.3.1. Associativity Instance

`Semigroup.to_isAssociative`: Semigroups satisfy `Std.Associative`

1.3.2. Function Composition

`comp_mul_left`: $(x \cdot) \circ (y \cdot) = ((x * y) \cdot)$

- Left multiplication by y then x equals left multiplication by $x * y$

`comp_mul_right`: $(\cdot\ x) \circ (\cdot\ y) = (\cdot\ (y * x))$

- Right multiplication by y then x equals right multiplication by $y * x$

1.4. Commutative Semigroup

1.4.1. Commutativity Variations

`mul_left_comm`: $a * (b * c) = b * (a * c)$

- Middle element can slide left

`mul_right_comm`: $a * b * c = a * c * b$

- Right two elements can swap

`mul_mul_mul_comm`: $(a * b) * (c * d) = (a * c) * (b * d)$

- Parallel multiplication

`mul_rotate`, `mul_rotate'`: Cyclic permutations

- $a * b * c = b * c * a$
- $a * (b * c) = b * (c * a)$

1.5. Monoid Properties

1.5.1. Identity Functions

`one_mul_eq_id`: Left multiplication by 1 is identity function

`mul_one_eq_id`: Right multiplication by 1 is identity function

1.5.2. Conditional Identities

`ite_mul_one`: `ite P (a * b) 1 = ite P a 1 * ite P b 1`

- Product of conditionals equals conditional of product `eq_one_iff_eq_one_of_mul_eq_one`: If $a * b = 1$, then $a = 1 \leftrightarrow b = 1$

1.5.3. Powers with Natural Numbers

`pow_boole`: $a^{\text{if } P \text{ then } 1 \text{ else } 0} = \text{if } P \text{ then } a \text{ else } 1$

`pow_mul_pow_sub`: For $m \leq n$: $a^m * a^{\{n-m\}} = a^n$

`pow_sub_mul_pow`: For $m \leq n$: $a^{\{n-m\}} * a^m = a^n$

1.6. Implementation Notes

- Heavy use of `@[to_additive]` to generate additive versions automatically
- Most proofs are one-liners using `simp`, `rw`, or definitional equality
- Systematic naming: multiplicative version first, then additive via attribute