# Equational theories

Contributors of the Equational Theory Project

October 3, 2024

# Chapter 1

# Basic theory of magmas

**Definition 1.1** (Magma)**.** A *magma* is a set $G$ equipped with a binary operation $\diamond : G \times G \to G$. A *homomorphism* $\varphi : G \to H$ between two magmas is a map such that $\varphi(x \diamond y) = \varphi(x) \diamond \varphi(y)$ for all $x, y \in G$. An *isomorphism* is an invertible homomorphism.

Groups, semi-groups, and monoids are familiar examples of magmas. However, in general we do not expect magmas to have any associative properties. In some literature, magmas are also known as groupoids, although this term is also used for a slightly different object (a category with inverses).

A magma is called *empty* if it has cardinality zero, *singleton* if it has cardinality one, and *non-trivial* otherwise.

The number of magma structures on a set $G$ of cardinality $n$ is of course $n^{n^2}$, which is [1]

$$1, 1, 16, 19683, 4294967296, 298023223876953125, ...$$

(OEIS A002489). Up to isomorphism, the number of finite magmas of cardinality $n$ up to isomorphism is the slightly slower growing sequence

$$1, 1, 10, 3330, 178981952, 2483527537094825, 1432559000331889152227568 0, ...$$

(OEIS A001329).

**Definition 1.2** (Free Magma)**.** The *free magma* $M_X$ generated by a set $X$ (which we call an *alphabet*) is the set of all finite formal expressions built from elements of $X$ and the operation $\diamond$. An element of $M_X$ will be called a *word* with alphabet $X$. The *order* of a word is the number of $\diamond$ symbols needed to generate the word. Thus for instance $X$ is precisely the set of words of order 0 in $M_X$.

For sake of concreteness, we will take the alphabet $X$ to default to the natural numbers $\mathbb{N}$ if not otherwise specified.

For instance, if $X = \{0, 1\}$, then $M_X$ would consist of the following words:

- 0, 1 (the words of order 0);
- $0 \diamond 0$, $0 \diamond 1$, $1 \diamond 0$, $1 \diamond 1$ (the words of order 1);

[1] All sequences start from $n = 0$ unless otherwise specified.

- $0 \diamond (0 \diamond 0)$, $0 \diamond (0 \diamond 1)$, $0 \diamond (1 \diamond 0)$, $0 \diamond (1 \diamond 1)$, $1 \diamond (0 \diamond 0)$, $1 \diamond (0 \diamond 1)$, $1 \diamond (1 \diamond 0)$, $1 \diamond (1 \diamond 1)$, $(0 \diamond 0) \diamond 0$, $(0 \diamond 0) \diamond 1$, $(0 \diamond 1) \diamond 0$, $(0 \diamond 1) \diamond 1$, $(1 \diamond 0) \diamond 0$, $(1 \diamond 0) \diamond 1$, $(1 \diamond 1) \diamond 0$, $(1 \diamond 1) \diamond 1$ (the words of order 2);
- etc.

**Lemma 1.3.** *For a finite alphabet $X$, the number of words of order $n$ is $C_n|X|^{n+1}$, where $C_n$ is the $n^{\text{th}}$ Catalan number and $X$ is the cardinality of $X$.*

*Proof.* Follows from standard properties of Catalan numbers. □

The first few Catalan numbers are

$$1, 1, 2, 5, 14, 42, 132, ...$$

(OEIS A000108).

**Definition 1.4** (Induced homomorphism)**.** Given a function $f : X \to G$ from an alphabet $X$ to a magma $G$, the *induced homomorphism* $\varphi_f : M_X \to G$ is the unique extension of $f$ to a magma homomorphism. Similarly, if $\pi : X \to Y$ is a function, we write $\pi_* : M_X \to M_Y$ for the unique extension of $\pi$ to a magma homomorphism.

For instance, if $f : \{0, 1\} \to G$ maps $0, 1$ to $x, y$ respectively, then

$$\varphi_f(0 \diamond 1) = x \diamond y$$

$$\varphi_f(1 \diamond (0 \diamond 1)) = y \diamond (x \diamond y)$$

and so forth. If $\pi : \mathbb{N} \to \mathbb{N}$ is the map $\pi(n) := n + 1$, then

$$\pi_*(0 \diamond 1) = 1 \diamond 2$$

$$\pi_*(1 \diamond (0 \diamond 1)) = 2 \diamond (1 \diamond 2)$$

and so forth.

**Definition 1.5** (Law)**.** Let $X$ be a set. A *law* with alphabet $X$ is a formal expression of the form $w \simeq w'$, where $w, w' \in M_X$ are words with alphabet $X$ (thus one can identify laws with alphabet $X$ with elements of $M_X \times M_X$). A magma $G$ *satisfies* the law $w \simeq w'$ if we have $\varphi_f(w) = \varphi_f(w')$ for all $f : X \to G$, in which case we write $G \models w \simeq w'$.

Thus, for instance, the commutative law

$$0 \diamond 1 \simeq 1 \diamond 0 \tag{1.1}$$

is satisfied by a magma $G$ if and only if

$$x \diamond y = y \diamond x \tag{1.2}$$

for all $x, y \in G$. We refer to (1.2) as the *equation* associated to the law (1.1). One can think of equations as the "semantic" interpretation of a "syntactic" law. However, we shall often abuse notation and a law with its associated equation thus we shall (somewhat carelessly) also refer to (1.2) as "the commutative law" (rather than "the commutative equation").

**Definition 1.6** (Models)**.** Given an arbitrary set $\Gamma$ of laws, a magma $G$ is a *model* of $\Gamma$ with the (overloaded) notation $G \models \Gamma$ if $G \models w \simeq w'$ for every $w \simeq w'$ in $\Gamma$; we also say that $G$ *obeys* $\Gamma$. Given a law $E$, we write $\Gamma \models E$ if every magma $G$ that models $\Gamma$, also models $E$.

**Definition 1.7** (Derivation)**.** Given a set $\Gamma$ of laws and a law $w \simeq w'$ over a fixed alphabet $X$, we say that $\Gamma$ *derives* $w \simeq w'$, and write $\Gamma \vdash w \simeq w'$, if the law can be obtained using a finite number of applications of the following rules:

1. if $w \simeq w' \in \Gamma$, then $\Gamma \vdash w \simeq w'$.
2. $\Gamma \vdash w \simeq w$ for any word $w$.
3. if $\Gamma \vdash w \simeq w'$ then $\Gamma \vdash w' \simeq w$.
4. if $\Gamma \vdash w \simeq w'$ and $\Gamma \vdash w' \simeq w''$ then $\Gamma \vdash w \simeq w''$.
5. if $\Gamma \vdash w \simeq w'$ then $\Gamma \vdash w\sigma \simeq w'\sigma$, where $\sigma$ is an arbitrary map from $X$ to words in $M_X$ and $w\sigma$ replaces each occurrence of an element of $X$ with it's image by $\sigma$ in $w$.
6. if $\Gamma \vdash w_1 \simeq w_2$ and $\Gamma \vdash w_3 \simeq w_4$ then $\Gamma \vdash w_1 \diamond w_3 \simeq w2 \diamond w_4$

This definition is useful because of the following theorem:

**Theorem 1.8** (Birkhoff's completeness theorem)**.** *For any set of laws $\Gamma$ and words $w, w'$ over a fixed alphabet*

$$\Gamma \vdash w \simeq w' \text{ iff } \Gamma \vDash w \simeq w'.$$

*Proof.* (Sketch) The 'only if' component is soundness, and follows from verifying that the rules of inference in Definition 1.7 holds for $\vDash$. The 'if' part is completeness, and is proven by constructing the magma of words, quotiented out by the relation $\Gamma \vdash w \simeq w'$, which is easily seen to be an equivalence relation respecting the magma operation. □

**Corollary 1.9** (Compactness theorem)**.** *Let $\Gamma$ be a collection of laws, and let $E$ be a law. Then $\Gamma \vDash E$ if and only if there exists a finite subset $\Gamma'$ of $\Gamma$ such that $\Gamma' \vDash E$.*

*Proof.* The claim is obvious for $\vdash$, and the claim then follows from Theorem 1.8. □

**Lemma 1.10** (Pushforward)**.** *Let $w \simeq w'$ be a law with some alphabet $X$, $G$ be a magma, and $\pi : X \to Y$ be a function. If $G \vDash w \simeq w'$, then $G \vDash \pi_*(w) \simeq \pi_*(w')$. In particular, if $\pi$ is a bijection, the statements If $G \vDash w \simeq w'$, then $G \vDash \pi_*(w) \simeq \pi_*(w')$ are equivalent.*

*Proof.* Trivial. □

If $\pi$ is a bijection, we will call $\pi_*(w) \simeq \pi_*(w')$ a *relabeling* of the law $w \simeq w'$. Thus for instance

$$5 \diamond 7 \simeq 7 \diamond 5$$

is a relabeling of the commutative law (1.1). By the above lemma, relabeling does not affect whether a given magna satisfies a given law.

**Lemma 1.11** (Equivalence)**.** *Let $G$ be a magma and $X$ be an alphabet. Then the relation $G \vDash w \simeq w'$ is an equivalence relation on $M_X$.*

*Proof.* Trivial. □

Define the total order of a law $w \simeq w'$ to be the sum of the orders of $w$ and $w'$.

**Lemma 1.12** (Counting laws up to relabeling)**.** *Up to relabeling, the number of laws $w \simeq w'$ of total order $n$ is $C_{n+1}B_{n+2}$.*

*Proof.* Follows from the properties of Catalan and Bell numbers. □

The first few Bell numbers are

$$1, 1, 2, 5, 15, 52, 203, ...$$

(OEIS A000110).

The sequence in Lemma 1.12 is

$$2, 10, 75, 728, 8526, 115764, ...$$

(OEIS A289679).

Now we would also like to count laws up to relabeling and symmetry.

**Lemma 1.13** (Counting laws up to relabeling and symmetry)**.** *Up to relabeling and symmetry, the number of laws $w \simeq w'$ of total order $n$ is*

$$C_{n+1}B_{n+2}/2$$

*when $n$ is odd, and*

$$(C_{n+1}B_{n+2} + C_{n/2}(2D_{n+2} - B_{n+2}))/2$$

*when $n$ is even, where $D_n$ is the number of partitions of $[n]$ up to reflection.*

*Proof.* Elementary counting. □

The sequence $D_n$ is

$$1, 1, 2, 4, 11, 32, 117, ...$$

(OEIS A103293), and the sequence in Lemma 1.13 is

$$2, 5, 41, 364, 4294, 57882, 888440, ...$$

(OEIS A376620).

We can also identify all laws of the form $w \simeq w$ with the trivial law $0 \simeq 0$. The number of such laws of total order $n$ is zero if $n$ is odd, and $C_{n/2}B_{n/2+1}$ if $n$ is even. We conclude:

**Lemma 1.14** (Counting laws up to relabeling, symmetry, and triviality)**.** *Up to relabeling, symmetry, and triviality, the number of laws of total order $n$ is*

$$C_{n+1}B_{n+2}/2$$

*if $n$ is odd, 2 if $n = 0$, and*

$$(C_{n+1}B_{n+2} + C_{n/2}(2D_{n+2} - B_{n+2}))/2 - C_{n/2}B_{n/2+1}$$

*if $n \geq 2$ is even.*

*Proof.* Routine counting. □

This sequence is

$$2, 5, 39, 364, 4284, 57882, 888365, ...$$

(OEIS A376640).

In particular, up to relabeling, symmetry, and triviality, there are exactly 4694 laws of total order at most 4. A list can be found here. A script for generating them may be found here. The list is sorted first by the total number of operations, then by the number of operations on the LHS. Within each such class we define an order on expressions by lexical order on variables (ordered $x, y, z, w, u, v$). The equation are arranged to be minimal with respect to this sorting order, thus the LHS will be shorter than or equal than the RHS, and earlier in the lexical order if the LHS and RHS are of equal length.

# Chapter 2

# Selected laws

In this project we study the 4694 laws (up to symmetry and relabeling) of total order at most 4. Selected laws of interest are listed below, as well as in this file.

**Definition 2.1** (Equation 1)**.** Equation 1 is the law $0 \simeq 0$ (or the equation $x = x$).

This is the trivial law, satisfied by all magmas. It is self-dual.

**Definition 2.2** (Equation 2)**.** Equation 2 is the law $0 \simeq 1$ (or the equation $x = y$).

This is the singleton law, satisfied only by the empty and singleton magmas. It is self-dual.

**Definition 2.3** (Equation 3)**.** Equation 3 is the law $0 \simeq 0 \diamond 0$ (or the equation $x = x \diamond x$).

This is the idempotence law. It is self-dual.

**Definition 2.4** (Equation 4)**.** Equation 4 is the law $0 \simeq 0 \diamond 1$ (or the equation $x = x \diamond y$).

This is the left absorption law.

**Definition 2.5** (Equation 5)**.** Equation 5 is the law $0 \simeq 1 \diamond 0$ (or the equation $x = y \diamond x$).

This is the right absorption law (the dual of Definition 2.4).

**Definition 2.6** (Equation 6)**.** Equation 6 is the law $0 \simeq 1 \diamond 1$ (or the equation $x = y \diamond y$).

This law is equivalent to the singleton law.

**Definition 2.7** (Equation 7)**.** Equation 7 is the law $0 \simeq 1 \diamond 2$ (or the equation $x = y \diamond z$).

This law is equivalent to the singleton law.

**Definition 2.8** (Equation 8)**.** Equation 8 is the law $0 \simeq 0 \diamond (0 \diamond 0)$ (or the equation $x = x \diamond (x \diamond x)$).

**Definition 2.9** (Equation 14)**.** Equation 14 is the law $0 \simeq 1 \diamond (0 \diamond 1)$ (or the equation $x = y \diamond (x \diamond y)$).

Appears in Problem A1 from Putnam 2001.

**Definition 2.10** (Equation 16)**.** Equation 16 is the law $0 \simeq 1 \diamond (1 \diamond 0)$ (or the equation $x = y \diamond (y \diamond x)$).

**Definition 2.11** (Equation 23)**.** Equation 23 is the law $0 \simeq (0 \diamond 0) \diamond 0$ (or the equation $x = (x \diamond x) \diamond x$).

This is the dual of Definition 2.8.

**Definition 2.12** (Equation 29)**.** Equation 29 is the law $0 \simeq (1 \diamond 0) \diamond 1$ (or the equation $x = (y \diamond x) \diamond y$).

Appears in Problem A1 from Putnam 2001. Dual to Definition 2.9.

**Definition 2.13** (Equation 38)**.** Equation 38 is the law $0 \diamond 0 \simeq 0 \diamond 1$ (or the equation $x \diamond x = x \diamond y$).

This law asserts that the magma operation is independent of the second argument.

**Definition 2.14** (Equation 39)**.** Equation 39 is the law $0 \diamond 0 \simeq 1 \diamond 0$ (or the equation $x \diamond x = y \diamond x$).

This law asserts that the magma operation is independent of the first argument (the dual of Definition 2.13).

**Definition 2.15** (Equation 40)**.** Equation 40 is the law $0 \diamond 0 \simeq 1 \diamond 1$ (or the equation $x \diamond x = y \diamond y$).

This law asserts that all squares are constant. It is self-dual.

**Definition 2.16** (Equation 41)**.** Equation 41 is the law $0 \diamond 0 \simeq 1 \diamond 2$ (or the equation $x \diamond x = y \diamond z$).

This law is equivalent to the constant law, Definition 2.20.

**Definition 2.17** (Equation 42)**.** Equation 42 is the law $0 \diamond 1 \simeq 0 \diamond 2$ (or the equation $x \diamond y = x \diamond z$).

Equivalent to Definition 2.13.

**Definition 2.18** (Equation 43)**.** Equation 43 is the law $0 \diamond 1 \simeq 1 \diamond 0$ (or the equation $x \diamond y = y \diamond x$).

The commutative law. It is self-dual.

**Definition 2.19** (Equation 45)**.** Equation 45 is the law $0 \diamond 1 \simeq 2 \diamond 1$ (or the equation $x \diamond y = z \diamond y$).

This is the dual of Definition 2.17.

**Definition 2.20** (Equation 46)**.** Equation 46 is the law $0 \diamond 1 \simeq 2 \diamond 3$ (or the equation $x \diamond y = z \diamond w$).

The constant law: all products are constant. It is self-dual.

**Definition 2.21** (Equation 168)**.** Equation 168 is the law $0 \simeq (1 \diamond 0) \diamond (0 \diamond 2)$ (or the equation $x = (y \diamond x) \diamond (x \diamond z)$).

The law of a central groupoid. It is self-dual.

**Definition 2.22** (Equation 381)**.** Equation 381 is the law $0 \diamond 1 \simeq (0 \diamond 2) \diamond 1$ (or the equation $x \diamond y = (x \diamond z) \diamond y$).

Appears in Putnam 1978, Problem A4, part (b).

**Definition 2.23** (Equation 387)**.** Equation 387 is the law $0 \diamond 1 \simeq (1 \diamond 1) \diamond 0$ (or the equation $x \diamond y = (y \diamond y) \diamond x$).

Introduced in MathOverflow.

**Definition 2.24** (Equation 953)**.** Equation 953 is the law $0 = 1 \diamond ((2 \diamond 0) \diamond (2 \diamond 2))$ (or the equation $x = y \diamond ((z \diamond x) \diamond (z \diamond z))$).

**Definition 2.25** (Equation 1571)**.** Equation 1571 is the law $0 \simeq (1 \diamond 2) \diamond (1 \diamond (0 \diamond 2))$ (or the equation $x = (y \diamond z) \diamond (y \diamond (x \diamond z))$).

Introduced in [5].

**Definition 2.26** (Equation 1689)**.** Equation 1689 is the law $0 \simeq (1 \diamond 0) \diamond ((0 \diamond 2) \diamond 2)$ (or the equation $x = (y \diamond x) \diamond ((x \diamond z) \diamond z)$).

Mentioned in [2].

**Definition 2.27** (Equation 2662)**.** Equation 2662 is the law $0 \simeq ((0 \diamond 1) \diamond (0 \diamond 1)) \diamond 0$ (or the equation $x = ((x \diamond y) \diamond (x \diamond y)) \diamond x$).

Appears in [5].

**Definition 2.28** (Equation 3722)**.** Equation 3722 is the law $0 \diamond 1 \simeq (0 \diamond 1) \diamond (0 \diamond 1)$ (or the equation $x \diamond y = (x \diamond y) \diamond (x \diamond y)$).

Appears in Putnam 1978, Problem A4, part (a). It is self-dual.

**Definition 2.29** (Equation 3744)**.** Equation 3744 is the law $0 \diamond 1 \simeq (0 \diamond 2) \diamond (3 \diamond 1)$ (or the equation $x \diamond y = (x \diamond z) \diamond (w \diamond y)$).

This law is called a "bypass operation" in Putnam 1978, Problem A4. It is self-dual.

**Definition 2.30** (Equation 4512)**.** Equation 4512 is the law $0 \diamond (1 \diamond 2) \simeq (0 \diamond 1) \diamond 2$ (or the equation $x \diamond (y \diamond z) = (x \diamond y) \diamond z$).

The associative law. It is self-dual.

**Definition 2.31** (Equation 4513)**.** Equation 4513 is the law $0 \diamond (1 \diamond 2) \simeq (0 \diamond 1) \diamond 3$ (or the equation $x \diamond (y \diamond z) = (x \diamond y) \diamond w$).

**Definition 2.32** (Equation 4522)**.** Equation 4522 is the law $0 \diamond (1 \diamond 2) \simeq (0 \diamond 3) \diamond 4$ (or the equation $x \diamond (y \diamond z) = (x \diamond w) \diamond u$).

Dual to Definition 2.34.

**Definition 2.33** (Equation 4564)**.** Equation 4564 is the law $0 \diamond (1 \diamond 2) \simeq (3 \diamond 1) \diamond 2$ (or the equation $x \diamond (y \diamond z) = (w \diamond y) \diamond z$).

Dual to Definition 2.31.

**Definition 2.34** (Equation 4579)**.** Equation 4579 is the law $0 \diamond (1 \diamond 2) \simeq (3 \diamond 4) \diamond 2$ (or the equation $x \diamond (y \diamond z) = (w \diamond u) \diamond z$).

Dual to Definition 2.32.

**Definition 2.35** (Equation 4582)**.** Equation 4582 is the law $0 \diamond (1 \diamond 2) \simeq (3 \diamond 4) \diamond 5$ (or the equation $x \diamond (y \diamond z) = (w \diamond u) \diamond v$).

This law asserts that all triple constants (regardless of bracketing) are constant.

# Chapter 3

# Infinite models

In this chapter we consider non-implications which are refuted only on infinite models, as those are more challenging to prove—they can't be proved by directly giving an operation table and checking which laws it satisfies.

We note some selected laws of order more than 5, used for such non-implications.

**Definition 3.1** (Equation 5105)**.** Equation 5105 is the law $0 \simeq 1 \diamond (1 \diamond (1 \diamond (0 \diamond (2 \diamond 1))))$ (or the equation $x = y \diamond (y \diamond (y \diamond (x \diamond (z \diamond y))))$).

This law of order 5 was mentioned in [2].

**Definition 3.2** (Equation 28393)**.** Equation 28393 is the law $0 \simeq (((0 \diamond 0) \diamond 0) \diamond 1) \diamond (0 \diamond 2)$ (or the equation $x = (((x \diamond x) \diamond x) \diamond y) \diamond (x \diamond z)$).

This law of order 5 was introduced by Kisielewicz [3].

**Definition 3.3** (Equation 374794)**.** Equation 374794 is the law $0 \simeq (((1 \diamond 1) \diamond 1) \diamond 0) \diamond ((1 \diamond 1) \diamond 2)$ (or the equation $x = (((y \diamond y) \diamond y) \diamond x) \diamond ((y \diamond y) \diamond z)$).

This law of order 6 was introduced by Kisielewicz [3].

The singleton or empty magma obeys all equational laws. One can ask whether an equational law admits nontrivial finite or infinite models. An *Austin law* is a law which admits infinite models, but no nontrivial finite models. Austin [1] established the first such law, namely the order 9 law

$$(((1 \diamond 1) \diamond 1) \diamond 0) \diamond (((1 \diamond 1) \diamond ((1 \diamond 1) \diamond 1)) \diamond 2) \simeq 0.$$

A shorter Austin law of order 6 was established in [3]:

**Theorem 3.4** (Kisielewicz's first Austin law)**.** *Definition 3.3 is an Austin law.*

*Proof.* Suppose for contradiction that we have a non-trivial model of Definition 3.3. Write $y^2 := y \diamond y$ and $y^3 := y^2 \diamond y$. For any $y, z$, introduce the functions $f_y : x \mapsto y^3 \diamond x$ and $g_{yz} : x \mapsto x \diamond (y^2 \diamond z)$. Definition 3.3 says that $g_{yz}$ is a left-inverse of $f_y$, hence by finiteness these are inverses and $g_{yz}$ is independent of $z$. In particular

$$f(y^3) = g_{yy}(y^3) = g_{yz}(y^3) = f(y^2 \diamond z)$$

and hence $y^2 \diamond z$ is independent of $z$. Thus

$$f_y(x) = (y^2 \diamond y) \diamond x = (y^2 \diamond y^2) \diamond x$$

is independent of $x$. As $f_y$ is invertible, this forces the magma to be trivial, a contradiction.

To construct an infinite magma, take the positive integers $\mathbb{Z}^+$ with the operation $x \diamond y$ defined as

- $2^x$ if $y = x$;
- $3^y$ if $x = 1 \neq y$;
- $\min(j, 1)$ if $x = 3^j$ and $y \neq x$; and
- 1 otherwise.

Then $y^2 = 2^y$, $y^3 = 1$, and $y^2 \diamond z$ a power of two for all $y, z$, and $(1 \diamond x) \diamond w = x$ for all $x$ whenever $w$ is a power of two, so Definition 3.3 is satisfied. □

An even shorter law (order 5) was obtained by the same author in a followup paper [2]:

**Theorem 3.5** (Kisielewicz theoremII)**.** *Definition 3.2 is an Austin law.*

*Proof.* Using the $y^2$ and $y^3$ notation as before, the law reads

$$x = (y^3 \diamond x) \diamond (y \diamond z). \tag{3.1}$$

In particular, for any $y$, the map $T_y \colon x \mapsto y^3 \diamond x$ is injective, hence bijective in a finite model $G$. In particular we can find a function $f : G \to G$ such that $T_y f(y) = y^3$ for all $y$ Applying (3.1) with $x = f(y)$, we conclude

$$T_y(y \diamond z) = y^3 \diamond (y \diamond z) = f(y)$$

and thus $y \diamond z$ is independent of $z$ by injectivity of $T_y$. Thus, the left-hand side of (3.1) does not depend on $x$, and so the model is trivial. This shows there are no non-trivial finite models.

To establish an infinite model, use $\mathbb{N}$ with $x \diamond y$ defined by requiring

$$y \diamond y = 2^y; \quad 2^y \diamond y = 3^y$$

and

$$3^y \diamond x = 3^y 5^x$$

for $x \neq 3^y$, and

$$(3^y 5^x) \diamond z = x$$

for $z \neq 3^y 5^x$. Finally set

$$2^{3^y} \diamond z = 3^y$$

for $z \neq 3^y, 2^{3^y}$. All other assignments of $\diamond$ may be made arbitrarily. It is then a routine matter to establish (3.1). □

In that paper a computer search was also used to show that no law of order four or less is an Austin law.

An open question is whether Definition 3.1 is an Austin law. We have the following partial result from [2]:

**Theorem 3.6** (Equation 5105 has no non-trivial finite models)**.** *Definition 3.1 has no non-trivial finite models.*

*Proof.* From Definition 3.1 we see that the map $w \mapsto y \diamond w$ is onto, hence injective in a finite model. Using this injectivity four times in Definition 3.1, we see that $z \diamond y$ does not depend on $z$, hence the expression $x \diamond (z \diamond y)$ does not depend on $x$. By Definition 3.1 again, this means that $x$ does not depend on $x$, which is absurd in a non-trivial model. □

We also have such a non-implication involving two laws of order 4:

**Definition 3.7** (Equation 3994)**.** Equation 3994 is the law $0 \diamond 1 \simeq (2 \diamond (0 \diamond 1)) \diamond 2$ (or the equation $x \diamond y = (z \diamond (x \diamond y)) \diamond z$).

**Definition 3.8** (Equation 3588)**.** Equation 3588 is the law $0 \diamond 1 \simeq 2 \diamond ((0 \diamond 1) \diamond 2)$ (or the equation $x \diamond y = z \diamond ((x \diamond y) \diamond z)$).

**Theorem 3.9.** *All finite magmas which satisfy definition 3.7 also satisfy definition 3.8*

*Proof.* For a finite magma $M$, consider the set $S = \{x \diamond y | x, y \in M\}$. Now $f_z : x \mapsto z \diamond x$ and $g_z : x \mapsto x \diamond z$. They both map $S$ to $S$, and due to the hypothesis $g_z \diamond f_z$ is the identity on $S$, so because $S$ is finite $f_z$ and $g_z$ must be inverse bijections on it, and therefore they commute. □

**Theorem 3.10.** *There exists a magma which satisfies 3.7 and not 3.8.*

*Proof.* Consider $\mathbb{N}$, with $x \diamond y$ defined as $x \oplus y$ (bitwise XOR) if $x$ and $y$ are even, $y+2$ if only $y$ is even, $x \dot{-} 2$ if only $x$ is even, and 0 if both are odd. Note that the range of the operation is the set of even naturals. Definition 3.7 is satisfied, because for even $z$ we get $z \oplus (x \diamond y) \oplus z = x \diamond y$ and for odd $z$ we get $(x \diamond y) + 2 \dot{-} 2 = x \diamond y$. Setting $x = y = z = 1$, definition 3.8 isn't satisfied. □

# Chapter 4

# General implications

We will be interested in seeing which laws imply which other laws, in the sense that magmas obeying the former law automatically obey the latter. We will also be interested in *anti-implications* showing that one law does *not* imply another, by producing examples of magmas that obey the former law but not the latter. Here is a formal definition.

**Definition 4.1** (Implication)**.** A law $E$ is said to *imply* another law $E'$ if $\{E\} \vDash E'$, or equivalently:

$$G \vDash w \simeq w' \implies G \vDash w'' \simeq w''' \text{ for all magmas } G$$

Two laws are said to be *equivalent* if they imply each other.

**Lemma 4.2** (Pre-order)**.** *If we define $E \leq E'$ if $E$ implies $E'$, then this is a pre-order on the set of laws, and equivalence is an equivalence relation.*

Note that we view the stronger law as less than or equal to the weaker law. This is because the class of magmas that obey the stronger law is a subset of the class of magmas that obey the weaker law. It is also consistent with the conventions of Lean's Mathlib.

*Proof.* Trivial. □

Implications between the laws from Chapter 2 are depicted in Figure 4.1.

**Lemma 4.3** (Maximal element)**.** *The law $0 \simeq 0$ is the maximal element in this pre-order.*

*Proof.* Trivial. □

**Lemma 4.4** (Minimal element)**.** *The law $0 \simeq 1$ is the minimal element in this pre-order.*

*Proof.* Trivial. □

Every magma $G$ has a *reversal* $G^{\mathrm{op}}$, formed by by replacing the magma operation $\diamond$ with its opposite $\diamond^{\mathrm{op}} : (x, y) \mapsto y \diamond x$. There is a natural isomorphism between these magmas, which induces an involution $w \mapsto w^{\mathrm{op}}$ on words $w \in M_X$. Every law $w \simeq w'$ then has a *dual* $w^{\mathrm{op}} \simeq (w')^{\mathrm{op}}$.

For instance, the dual of the law $0 \diamond 1 = 0 \diamond 2$ is $1 \diamond 0 = 2 \diamond 0$, which after relabeling is $0 \diamond 1 = 2 \diamond 1$. A list of equations and their duals can be found here. Of the 4694 equations under consideration, 84 are self-dual, leaving 2305 pairs of dual equations.

The pre-ordering on laws has a duality symmetry:

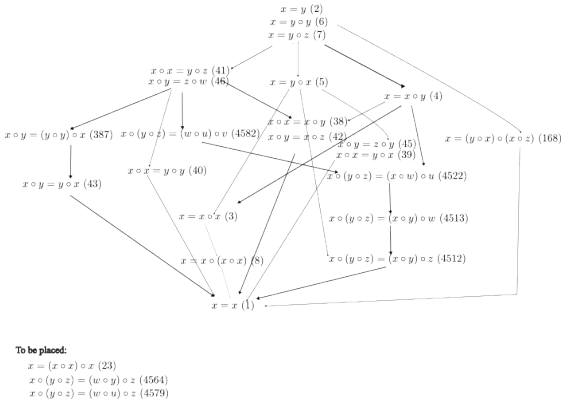


Figure 4.1: Implications between the above equations, displayed as a Hasse diagram.

**Lemma 4.5** (Duality of laws)**.** *If $w \simeq w'$ implies $w'' \simeq w'''$, then $w^{\mathrm{op}} \simeq (w')^{\mathrm{op}}$ implies $w''^{\mathrm{op}} \simeq (w''')^{\mathrm{op}}$.*

*Proof.* This follows from the fact that a magma $G$ satisfies a law $w \simeq w'$ if and only if $G^{\mathrm{op}}$ satisfies $w^{\mathrm{op}} \simeq (w')^{\mathrm{op}}$. □

Some equational laws can be "diagonalized":

**Theorem 4.6** (Diagonalization)**.** *An equational law of the form*

$$F(x_1, \dots, x_n) = G(y_1, \dots, y_m), \tag{4.1}$$

*where $x_1, \dots, x_n$ and $y_1, \dots, y_m$ are distinct elements of the alphabet, implies the diagonalized law*

$$F(x_1, \dots, x_n) = F(x'_1, \dots, x'_n).$$

*where $x'_1, \dots, x'_n$ are distinct from $x_1, \dots, x_n$ In particular, if $G(y_1, \dots, y_m)$ can be viewed as a specialization of $F(x'_1, \dots, x'_n)$, then these two laws are equivalent.*

*Proof.* From two applications of (4.1) one has

$$F(x_1, \dots, x_n) = G(y_1, \dots, y_m)$$

and

$$F(x'_1, \dots, x'_n) = G(y_1, \dots, y_m)$$

whence the claim. □

Thus for instance, Definition 2.7 is equivalent to Definition 2.2.

**Theorem 4.7** (Laws implied by the constant law)**.** *If $w, w'$ each have order at least one, then the law $w \simeq w'$ is implied by the constant law (Definition 2.20). If exactly one of $w, w'$ has order zero, and the law $w \simeq w'$ is not implied by the constant law.*

*Proof.* Routine. □

**Theorem 4.8** (Criterion for implication)**.** *If $w \simeq w'$ is such that every variable appears the same number of times in both $w$ and $w'$, and $w \simeq w'$ implies another law $w'' \simeq w'''$, then every variable appears the same number of times in both $w''$ and $w'''$.*

*Proof.* Consider the magma MS of multisets over an arbitrary set $A$ (which can be seen as finitely supported maps $A \to \mathbb{N}$), with the multiset addition law $+$. By hypothesis, this magma obeys $w \simeq w'$, and hence $w'' \simeq w'''$, giving the claim by comparing the orders of the elements of $A$ appearing in $w''$ and $w'''$ in this magma. □

# Chapter 5

# Implications between selected laws

We collect here some notable implications between the the selected laws in Chapter 2. By Theorem 1.8, every implication can basically be established by a finite number of rewrites. In most cases, the sequence of rewrites is quite straightforward, and the implication is very easy, but we record some less obvious examples.

**Theorem 5.1** (387 implies 43)**.** *Definition 2.23 implies Definition 2.18.*

*Proof.* (From MathOverflow). By Definition 2.23, one has the law

$$(x \diamond x) \diamond y = y \diamond x. \tag{5.1}$$

Specializing to $y = x \diamond x$, we conclude

$$(x \diamond x) \diamond (x \diamond x) = (x \diamond x) \diamond x$$

and hence by another application of (2.23) we see that $x \diamond x$ is idempotent:

$$(x \diamond x) \diamond (x \diamond x) = x \diamond x. \tag{5.2}$$

Now, replacing $x$ by $x \diamond x$ in (5.1) and then using (5.2) we see that

$$(x \diamond x) \diamond y = y \diamond (x \diamond x)$$

so in particular $x \diamond x$ commutes with $y \diamond y$:

$$(x \diamond x) \diamond (y \diamond y) = (y \diamond y) \diamond (x \diamond x). \tag{5.3}$$

Also, from two applications of (5.1) one has

$$(x \diamond x) \diamond (y \diamond y) = (y \diamond y) \diamond x = x \diamond y.$$

Thus (5.3) simplifies to $x \diamond y = y \diamond x$, which is Definition 2.18. □

**Theorem 5.2** (29 equivalent to 14)**.** *Definition 2.12 is equivalent to Definition 2.9.*

This result was posed as Problem A1 from Putnam 2001.

*Proof.* By Lemma 4.5 it suffices to show that Definition 2.12 implies Definition 2.9. From Definition 2.12 one has

$$x = ((x \diamond y) \diamond x) \diamond (x \diamond y)$$

and also

$$y = (x \diamond y) \diamond x$$

giving $x = y \diamond (x \diamond y)$, which is Definition 2.9. □

**Theorem 5.3** (14 implies 29)**.** *Definition 2.9 implies Definition 2.12.*

This result was posed as Problem A1 from Putnam 2001.

*Proof.* □

The following result was Problem A4 on Putnam 1978.

**Theorem 5.4** (3744 implies 3722, 381)**.** *Definition 2.29 implies Definition 2.28 and Definition 2.22.*

*Proof.* By hypothesis, one has

$$x \diamond y = (x \diamond z) \diamond (w \diamond y)$$

for all $x, y, z, w$. Various specializations of this give

$$x \diamond y = (x \diamond z) \diamond (y \diamond y) \tag{5.4}$$
$$x \diamond z = (x \diamond z) \diamond (x \diamond z) \tag{5.5}$$
$$(x \diamond z) \diamond y = ((x \diamond z) \diamond (x \diamond z)) \diamond (y \diamond y). \tag{5.6}$$

The equation (5.5) gives Definition 2.28, while (5.4), (5.5), (5.6) gives

$$x \diamond y = (x \diamond z) \diamond y$$

which is Definition 2.22. □

**Theorem 5.5** (1689 is equivalent to 2)**.** *Definition 2.26 is equivalent to Definition 2.2.*

*Proof.* The implication of Definition 2.26 from Definition 2.2 is trivial. The converse is a surprisingly long chain of implications; see pages 326–327 of [2]. The initial law

$$x = (y \diamond x) \diamond ((x \diamond z) \diamond z)$$

is used to obtain, in turn,

$$x \diamond ((((x \diamond y) \diamond y) \diamond z) \diamond z) = (x \diamond y) \diamond y,$$
$$(x \diamond (y \diamond z)) \diamond (z \diamond ((z \diamond w) \diamond w)) = y \diamond z,$$
$$x \diamond (y \diamond ((y \diamond z) \diamond z)) = (x \diamond y) \diamond y,$$
$$((x \diamond (y \diamond z)) \diamond z) \diamond z = y \diamond z,$$
$$(x \diamond (y \diamond (z \diamond w))) \diamond (z \diamond w) = y \diamond (z \diamond w),$$
$$(x \diamond (y \diamond z)) \diamond (y \diamond z) = x \diamond (y \diamond z),$$
$$((x \diamond y) \diamond ((y \diamond z) \diamond z)) \diamond ((y \diamond z) \diamond z) = y,$$

$$((x \diamond y) \diamond ((y \diamond z) \diamond z)) \diamond ((y \diamond z) \diamond z) = ((x \diamond ((x \diamond y) \diamond ((y \diamond z) \diamond z))) \diamond ((y \diamond z) \diamond z)) \diamond ((y \diamond z) \diamond z),$$

$$x \diamond ((x \diamond y) \diamond y) = x,$$

$$x \diamond (x \diamond (y \diamond z)) = x,$$

$$(x \diamond y) \diamond y = x \diamond y,$$

$$(x \diamond x) \diamond x = x,$$

$$(x \diamond y) \diamond y = y,$$

$$x \diamond y = y.$$

□

The following result was established in [5].

**Theorem 5.6** (Consequences of 1571)**.** *Magmas obeying Definition 2.25 also obey Definitions 2.27, 2.15, 2.11, 2.8, 2.10,* (2.9)*, 2.18, and 2.30, and are in fact abelian groups of exponent two. Conversely, all abelian groups of exponent two obey Definition 2.25.*

*Proof.* Suppose that a magma $G$ obeys Definition 2.25, thus

$$x = (y \diamond z) \diamond (y \diamond (x \diamond z)). \tag{5.7}$$

$$x = ((x \diamond y) \diamond (x \diamond y)) \diamond ((x \diamond y) \diamond (x \diamond (x \diamond y)))$$

and

$$x = (x \diamond y) \diamond (x \diamond (x \diamond y))$$

whence

$$x = ((x \diamond y) \diamond (x \diamond y)) \diamond x$$

which is Definition 2.27. This gives

$$y = ((y \diamond z) \diamond (y \diamond z)) \diamond y$$

while from (5.7) one has

$$(y \diamond z) \diamond (y \diamond z) = (x \diamond y) \diamond (x \diamond ((y \diamond z) \diamond (y \diamond z) \diamond y))$$

whence

$$(x \diamond y) \diamond (x \diamond y) = (y \diamond z) \diamond (y \diamond z).$$

This implies that $(x \diamond y) \diamond (x \diamond y)$ does not depend on $x$, or on $y$, hence is equal to some constant $e$:

$$(x \diamond y) \diamond (x \diamond y) = e.$$

From (5.7) the magma operation is surjective, hence

$$x \diamond x = e \tag{5.8}$$

which gives Definition 2.15. Applying (5.7) with $x = y = z$ we conclude

$$x = e \diamond (x \diamond e)$$

while if we instead take $y = z = e$ we have

$$x = e \diamond (e \diamond (x \diamond e))$$

hence

$$x = e \diamond x$$

and then also

$$x = x \diamond e$$

from which we readily conclude Definitions 2.11, 2.8; thus $e$ is an identity element. From (5.7) with $z = e$ we now have

$$x = y \diamond (y \diamond x) \tag{5.9}$$

which is Definition 2.10. If instead we take $y = e$ we have

$$x = z \diamond (x \diamond z) \tag{5.10}$$

which is Definition 2.9. So if we substitute $z = x \diamond y$ and use (5.9) we obtain

$$x = (x \diamond y) \diamond y$$

and hence

$$y \diamond x = y \diamond ((x \diamond y) \diamond y) = x \diamond y$$

thanks to (5.10). This gives Definition 2.18, thus $G$ is now commutative. From (5.7) once more one has

$$x \diamond (y \diamond z) = (y \diamond x) \diamond (z \diamond ((x \diamond (y \diamond z)) \diamond x))$$

which one can simplify using commutativity and (5.9) (or (5.10)) to eventually obtain

$$x \diamond (y \diamond z) = (x \diamond y) \diamond z$$

which is Definition 2.30. $G$ is now commutative and associative, and every element is its own inverse and of exponent 2, hence is an abelian group thanks to (5.8), so $G$ is an abelian group of exponent 2 as claimed. The converse is easily verified. □

**Theorem 5.7** (953 is equivalent to 2)**.** *Definition 2.24 is equivalent to Definition 2.2.*

*Proof.* It suffices to show that Definition 2.24 implies Definition 2.2. Pick an element 0 of $G$ and define $1 = 0 \diamond 0$ and $2 = 1 \diamond 1$ (we do not require $0, 1, 2$ to be distinct). From Definition 2.24 with $x = z = 0$ we have

$$0 = y \diamond 2.$$

If we then apply Definition 2.24 with $z = 1$ we conclude that

$$x = y \diamond 0$$

for all $x, y$, from which one concludes $x = x'$ for any $x, x' \in G$, giving Definition 2.2. □

# Chapter 6

# Selected magmas

Each magma can be used to establish anti-implications: if $\Gamma$ is the set of all laws obeyed by a magma $G$, then we have $\neg E \leq E'$ whenever $E \in \Gamma$ and $E' \notin \Gamma$. Large numbers of implications can already be obtained from

- All magmas of order at most 4, up to isomorphism (of which there are $178,985,294$);
- All commutative magmas of order 5, up to isomorphism **determine their count**;
- Cyclic groups $\mathbb{Z}/N\mathbb{Z}$ with $2 \leq N \leq 12$ and $x \circ y = ax^2 + bxy + cy^2 + dx + ey$ for randomly chosen $a, b, c, d, e$.
- There are only 1410 distinct cancellative magmas of order 5 (up to isomorphism), and Mace4 can generate all of them in under 20 seconds. A shell script to do this is available here. A magma is cancellative if $xy = xz$ implies $y = z$ and $yx = zx$ implies $y = z$.

Some other magmas have been used to establish counterexamples:

- The cyclic group $\mathbb{Z}/6\mathbb{Z}$ with the addition law.
- The natural numbers with law $x \circ y = x + 1$.
- The natural numbers with law $x \circ y = xy + 1$.
- The reals with $x \circ y = (x + y)/2$.
- The natural numbers with $x \circ x$ equal to $x$ when $x = y$ and $x + 1$ otherwise.
- The set of strings with $x \circ y$ equal to $y$ when $x = y$ or when $x$ ends with $yyy$, or $xy$ otherwise (see this Zulip thread).
- The set of strings with $x \cdot y$ equal to $a$ if $x = aaaa$, and $xy$ otherwise. (See previous thread.)
- Vector spaces $\mathbb{F}_2^n$ over $\mathbb{F}_2$, which obey Definition 2.25 (and hence all the subsequent laws mentioned in Theorem 5.6).

# Chapter 7

# Equivalence with the constant and singleton laws

85 laws have been shown to be equivalent to the constant law (Definition 2.20), and 815 laws have been shown to be equivalent to the singleton law (Definition 2.2).

These are the laws up to 4 operations that follow from diagonalization of 2.2 and of 2.20.

In order to formalize these in Lean, a search was run on the list of equations to discover diagonalizations of these two specific laws: equations of the form $x = R$ where $R$ doesn't include $x$, and equations of the form $x \circ y = R$ where R doesn't include $x$ or $y$.

The proofs themselves all look alike, and correspond exactly to the two steps described in the proof of 4.6. The Lean proofs were generated semi-manually, using search-and-replace starting from the output of `grep` that found the diagonalized laws.

In the case of the constant law, equation 2.16 ($x \circ x = y \circ z$) wasn't detected using this method. It was added manually to the file with the existing proof from the sub-graph project.

# Chapter 8

# Metatheorems from Invariants

For the purposes of this chapter, a *theorem* is a (true) statement about particular equations, for example '(387 implies 43)' is a theorem. A *metatheorem* is a general statement about implications; one can usually get many theorems from a single metatheorem. This chapter is all about generating many interesting metatheorems using a *meta-metatheorem*, called the fundamental property of invariants. If all this is making your head spin, don't worry. Look at the sections below for examples of metatheorems you can probably agree are both concrete and interesting. Once you have done that, come back here and we will show you how to prove these and other metatheorems using *invariants*.

## 8.1 Invariants

Let $E, E_1$, and $E_2$ be equations. If $E \Rightarrow E_1$ and $E_1 \Rightarrow E_2$, then $E \Rightarrow E_2$. Very trivial. Rephrasing this, we see that if $E \Rightarrow E_1$ and $E \not\Rightarrow E_2$, then $E_1 \not\Rightarrow E_2$.

Extending this idea, suppose we compute the set of all equations which are implied by $E$; we will call this set $\mathcal{Y}(E)$ (we use $\mathcal{Y}$ because this is an example of a Yoneda embedding). Then $\mathcal{Y}(E)$ is upwards closed, or closed under forward implication: no equation in $\mathcal{Y}(E)$ can imply an equation not in $\mathcal{Y}(E)$. If we know $\mathcal{Y}(E)$ well, this already settles a potentially large number of implications in the negative.

While computing $\mathcal{Y}(E)$ for an arbitrary equation $E$ may seem daunting, for some nice equations we can find *invariants*, which makes the task manageable. An *invariant* for $E$ is some sort of data associated with expressions $w$ so that

$$\mathcal{Y}(E) = \{w = w' \mid \text{Invariant}(w) = \text{Invariant}(w')\}$$

If we can find an invariant which is computable for each term $w$, then we can easily describe $\mathcal{Y}(E)$. The fact that $\mathcal{Y}(E)$ is upwards closed is rephrased as follows; this is called **the fundamental property of invariants**. Remember that an invariant is a function taking expressions and outputting some data.

**Meta-metatheorem 8.1** (Fundamental property of invariants)**.** *Let $I$ be an invariant of $E$. If $w = w'$ implies $w'' = w'''$ and $I(w) = I(w')$ (that is, $E$ implies $w = w'$), then $I(w'') = I(w''')$.*

More succinctly, for an invariant $I$ of $E$ we must have

$$(w = w' \Rightarrow w'' = w''') \implies (I(w) = I(w') \Rightarrow I(w'') = I(w''')).$$

When using this result, we commonly take the contrapositive: if $I(w) = I(w')$ and $I(w'') \neq I(w''')$, then $w = w'$ cannot imply $w'' = w'''$. Note that the conclusion is independent of the equation $E$; all we need to know is that $I$ is an invariant.

*Note for category theorists.* Let $\Pi$ denote the preorder of magma equations ordered by implication. If $I$ is an invariant then define

$$I(w = w') := \begin{cases} \texttt{true} & \text{if } I(w) = I(w') \\ \texttt{false} & \text{otherwise} \end{cases}.$$

(In programming languages we would say $I(w = w') := I(w) == I(w')$). Let $\mathbf{Bool} = \{\texttt{true}, \texttt{false}\}$ be the poset where $\texttt{false} \leq \texttt{true}$. Then $I$ becomes a function $\Pi \to \mathbf{Bool}$, and the fundamental property of invariants just says that this function is monotone, i.e. functorial. Thus for every invariant $I$ we obtain a functor $\Pi \to \mathbf{Bool}$.

Question 1: Does every functor $\Pi \to \mathbf{Bool}$ come from an invariant?

Question 2: What can we say about the category of functors $\Pi \to \mathbf{Bool}$? Give a nice interpretation of natural transformations between invariants. □

The fundamental property of invariants is not a theorem, nor a metatheorem: it is a meta-metatheorem, in the sense that it will allow us to get a metatheorem for every invariant we find.

**Example: absorption law**

Let $E$ be the equation $x \diamond y = x$. Intuitively, we must have

$$\mathcal{Y}(E) = \{w = w' \mid \text{the leftmost variable is the same for } w \text{ and } w'\}.$$

We will talk about proving statements like this one (say in Lean) later on; take it as given for now. The invariant is clear: we define $I(w)$ to be the leftmost variable of $w$. Instantiating this invariant in the fundamental property of invariants, we get the following metatheorem.

**Metatheorem 8.2.** *Let $w = w'$ be an equation such that the leftmost variable of $w$ is the same as the leftmost variable of $w'$. Then $w = w'$ cannot imply an equation that does not have the property from the last sentence.*

**Example: associativity**

For a more complicated example, let $E$ be the associativity equation $x \diamond (y \diamond z) = (x \diamond y) \diamond z$. Intuitively, we must have

$$\mathcal{Y}(E) = \{\text{equations that, when we remove all parentheses, are of the form } w = w\}.$$

There is an invariant lurking behind: it is the (ordered) list of variables appearing in an expression, counting repetitions. More formally, we define $I(w)$ to be the tuple of variables appearing in $w$, listed from left to right, say. Again, from the fundamental property of invariants we get the following.

**Metatheorem 8.3.** *Let $w = w'$ be an equation such that the variables appearing in $w$, taking into account order and repetitions, are the same ones that appear in $w'$. Then $w = w'$ cannot imply an equation that does not have the property from the last sentence.*

If we were coding a computer program that computes $I(w)$ given $w$, one could take the string of symbols that is $w$, ignore all parentheses, replace all symbols $\diamond$ by commas, and surround with an appropriate delimiter. (I imagine one could easily do this using regular expressions.

We can compute other examples, but the invariant can get complicated even for simple equations. Exercise: what is the invariant for commutativity? Answer: To compute $I(w)$ from $w$ replace all parentheses with curly braces and all symbols $\diamond$ with commas, and interpret the result as nested sets.

## 8.2 Expanding the language

The method of invariants really shines when we expand our formal language. Right now our language consists of variables, parentheses, the equal sign, and $\diamond$ (there is also an implicit use of $\forall$ but let's ignore that for now). Let $\Pi$ denote the preorder of equations (built from the language described) ordered by implication.

We will add the symbol $\wedge$ ('and') to our language. Then we consider a bigger preorder $\Pi' \supseteq \Pi$ which includes equations and also conjunctions of equations. Even if we only care about $\Pi$ it will be apparent that studying invariants in $\Pi'$ gives us useful metatheorems about $\Pi$. Equations and conjunctions of equations are examples of *formulas* (or formulae, according to taste).

If $\varphi$ is a formula, we can define $\mathcal{Y}(\varphi)$ to be the set of all formulae implied by $\varphi$; this agrees with our previous definition. Now define an invariant of $\varphi$ to be a function $I$ on terms such that

$$\mathcal{Y}(\varphi) \cap \Pi = \{w = w' \mid I(w) = I(w')\}.$$

Again, this clearly agrees with our previous definition. Although $\mathcal{Y}(\varphi) \cap \Pi$ might not be upwards closed in $\Pi'$, it is upwards closed in $\Pi$, which is enough to get the fundamental property of invariants *verbatim*. This leads to more metatheorems we didn't have access to before.

#### Example: associativity and idempotency

Let $\varphi$ be the conjunction of the associative law and the idempotency law ($x \diamond x = x$). Again, we will rely on our intuition, which says that an invariant $I$ defined by taking $I(w)$ to be the set of all variables appearing in $w$, works. The corresponding metatheorem is the following

**Metatheorem 8.4.** *Let $w = w'$ be an equation such that the set of variables appearing in $w$ is equal to the set of variables appearing on $w'$. Then $w = w'$ cannot imply an equation that does not have the property from the last sentence.*

#### Example: associativity and commutativity

For a similar example, we can let $\varphi$ be the conjunction of the associative and the commutative laws. Here we can define $I(w)$ to be the multiset of variables appearing in $w$. We obtain the following metatheorem.

**Metatheorem 8.5.** *Let $w = w'$ be an equation such that the variables appearing in $w$, taking into account multiplicity, are the same ones that appear in $w'$. Then $w = w'$ cannot imply an equation that does not have the property from the last sentence.*

Trivia: this was the first example of a metatheorem obtained by use of an invariant.

**Example: associativity and commutativity with a twist**

We can keep expanding our language if it helps us express more intricate invariants. For instance, we can add the symbol '1' to our language. Let $\varphi$ be the conjunction of associativity, commutativity, the equations $1 \diamond x = x$, and

$$\underbrace{x \diamond x \diamond \cdots \diamond x}_{m \text{ times}} = 1,$$

for some fixed positive integer $m$. Pause to guess the invariant before we move on.

The invariant $I(w)$ is the multiset of variables appearing in $w$ but multiplicities are computed modulo $m$. Thus we have the pretty metatheorem:

**Metatheorem 8.6.** *Fix some positive integer $m$. Let $w = w'$ be an equation such that every variable appearing in $w$ appears the same number of times in $w'$ modulo $m$. Then $w = w'$ cannot imply an equation that does not have the property from the last sentence.*

## 8.3 Proving metatheorems from invariants in Lean

For the rest of this chapter we readopt the convention of calling 'theorem' an important result, not necessarily pertaining to specific equations.

An invariant is generally a *syntactic* property of an expression. However, invariants can also be described and calculated *semantically* through the notion of a *lifting magma family*, described below. The general idea is that the value of an invariant for an expression can be computed by substituting specific values for the variables in the expression and evaluating the result in a certain magma in the lifting magma family; additional requirements ensure that the fundamental property of invariants is satisfied.

**Definition 8.7** (Lifting Magma Family)**.** A *lifting magma family* is a family of magmas $\{G_\alpha\}$, one for each type $\alpha$, satisfying the following properties:

- For each type $\alpha$, there is a function $\iota_\alpha : \alpha \to G_\alpha$.
- Given a function $f : \alpha \to G_\alpha$, there is a magma homomorphism lift $f : G_\alpha \to G_\alpha$ such that lift $f(\iota_\alpha(x)) = f(x)$ for all $x$ in $\alpha$.

**Example 1.** *The free Abelian groups form a lifting magma family. When the underlying set is finite, the groups are isomorphic to $\mathbb{Z}^n$.*

**Example 2.** *Lists form a lifting magma family.*

The key consequence of the definition 8.7 is that it is significantly easier to check whether an equation is satisfied in a lifting magma family.

**Theorem 8.8** (Evaluation theorem for lifting magma families)**.** *Suppose $E$ is an equation involving a set of variables $X$, and let $G$ be a lifting magma family.*

*Determining whether $E$ is satisfied by $G_X$ is equivalent to checking that $E$ is true with the specific substitution $\iota_X$.*

*Proof.* For the forward direction, suppose $E$ is satisfied by $G_X$. Then, by definition, any substitution of the variables in $E$ with elements of $G_X$ will yield a true equation. In particular, substituting according to $\iota_X$ will yield a true equation.

For the reverse direction, suppose that $E$ is true when evaluated with the substitution $\iota_X$. Now, consider an arbitrary substitution of variables $f : X \to G_X$. By the lifting magma family

property, there is a magma homomorphism $\operatorname{lift} f : G_X \to G_X$ such that $\operatorname{lift} f(\iota_X(x)) = f(x)$ for all $x$ in $X$. In other words, applying the substitution $f$ is equivalent to first applying to substitution $\iota_X$ and then applying the homomorphism $\operatorname{lift} f$. Since $E$ is true when evaluated with the substitution $\iota_X$, it is also true after applying the homomorphism $\operatorname{lift} f$. Thus, $E$ is satisfied by $G_X$. □

**Theorem 8.9** (The fundamental property of invariants)**.** *Let $E$ and $E'$ be equations involving a set of variables $X$, and let $G$ be a lifting magma family.*

*If $E$ is true with the substitution $\iota_X$, and $E$ implies $E'$, then so is $E'$.*

*Proof.* Applying the evaluation theorem 8.8, we see that $E$ is satisfied by $G_X$. Since $E$ implies $E'$, $E'$ is also satisfied by $G_X$, and in particular, $E'$ is true with the substitution $\iota_X$. □

**Remark 1.** *The result of evaluating an expression along the function $\iota_X : X \to G_X$ is the invariant.*

*In the case of Abelian groups, the result of evaluation is the variables in the expression with multiplicity. In the case of lists, the result of evaluation is the variables in the expression in the order they appear.*

*When the lifting magma family has good computational properties, calculating the invariant becomes easy.*

**Remark 2.** *Given an equation $\phi$ in the language of magmas (possibly involving logical operations other than equality and universal quantification), the initial (i.e., most general) magmas satisfying $\phi$ (provided they exist) form a lifting magma family.*

*However, for the purpose of generating invariants, we are interested in lifting magma families with convenient descriptions that are computationally tractable.*

**Remark 3.** *Suppose $S$ is a finite set of equations in the language of magmas that is a confluent term rewriting system under a certain ordering of the terms (in the sense of the Knuth-Bendix algorithm). Then the initial magmas satisfying $S$ form a lifting magma family where equality of elements in the magma is decidable.*

*This offers a way of generating examples of lifting magma families with good computational properties for computing invariants of expressions.*

## 8.4 Conclusion: Beyond Invariants

We are still lacking:

- A large collection of invariants.
- An estimate for how many implications the resulting metatheorems will settle.
- Algorithms (in Lean, Python, or otherwise) to compute known invariants.
- General results about lifting magmas.
- Formalization of the method of invariants and resulting metatheorems.
- Knowledge about the category-theoretic interpretation of invariants (see the questions in the note for category theorists).

Related to the last bullet point, we note the following. If all that matters about invariants is the fundamental property, we can apply the old French trick of turning a (meta-meta)theorem into a definition.

Q: If we were to define invariants as any functions satisfying the fundamental property, would anything change? (For those who read the note for category theorists: an equivalent redefinition is to consider invariants as functors $\Pi \to \mathbf{Bool}$).

# Chapter 9

# Simple rewrites

53,905 implications were automatically generated by simple rewrites.
**describe the process of automatically generating these implications here.**

# Chapter 10

# Trivial auto-generated theorems

Approximately 4.5m transitive implications were proven by a transitive reduction of about 15k theorems. Most of these implications were derived from being the first automated run to connect the largest equivalence classes, hence creating a large set of transitively closed implications.

Scripts generated theorems to try simple combinations of equation rewrites to reach the desired goal for every unknown implication. The generated proof scripts were run with lean and the successful theorems were extracted. An example of the types of generated rewrites that were tested:

```
repeat intro
apply

repeat intro
try { rw [<-h] }
try { rw [<-h, <-h] }
try { rw [<-h, <-h, <-h] }
try { rw [<-h, <-h, <-h, <-h] }
try { rw [<-h, <-h, <-h, <-h, <-h] }
repeat rw [h]

repeat intro
try {
  nth_rewrite 1 [h]
  try { rw [h] }
  try { rw [<-h] }
}
try {
  nth_rewrite 2 [h]
  try { rw [h] }
  try { rw [<-h] }
}
try {
  nth_rewrite 3 [h]
  try { rw [h] }
  try { rw [<-h] }
}
```

```
try {
  nth_rewrite 4 [h]
  try { rw [h] }
  try { rw [<-h] }
}
try {
  nth_rewrite 1 [h]
  nth_rewrite 1 [h]
  try { rw [h] }
  try { rw [<-h] }
}
...
```

# Chapter 11

# Enumerating Small Finite Magmas

**describe the process of automatically generating these implications here.**

# Chapter 12

# Equation Search

Approximately 650k transitive implications were proven by a custom tool leveraging the implication graph. After previous brute force had derived many implications expressible as a small number of rewrites, this search tool uses substitutions implied by the implication graph to search further.

An example proof illustrates the logic it uses:

```
have eq3315 (x y : G) : x * y = x * (y * (x * x)) := by
  apply Apply.Equation12_implies_Equation11 at h
  apply RewriteHypothesis.Equation11_implies_Equation3323 at h
  apply Apply.Equation3323_implies_Equation3315 at h
  apply h
have eq52 (x y : G) : x = x * (y * (x * x)) := by
  apply Apply.Equation12_implies_Equation61 at h
  apply Apply.Equation61_implies_Equation54 at h
  apply Apply.Equation54_implies_Equation52 at h
  apply h
repeat intro
nth_rewrite 1 [eq3315]
nth_rewrite 1 [← eq52]
apply h
repeat assumption
```

Using the graph of implications and refutations, it identifies equivalence classes/strongly-connected components in the implication graph and possible goals by subtracting out the refutation graph. Iterating through all equivalence classes, it can perform a meet-in-the-middle graph search where it searches outwards from both hypotheses and goals by performing equation substitutions. Depending on the number of hypotheses versus goals, it dynamically adjusts the search depth on both sides based on a configured branching factor.

Due to it's naive implementation, it may only be able to perform certain substitutions in a round-about way and the graph size explodes faster than it must, so it's limited to fairly shallow search depths. Also, the tool may emit proofs without some information Lean may require, so some generated proofs have to be fixed-up afterwards.

# Chapter 13

# E-Graphs

For proving implications, we used another technique called equality saturation [6] with the lean-egg tactic, to automatically construct proofs.

A similar approach is being pursued in the MagmaEgg tool as well, which is a standalone program that only supports magma equalities, while the lean-egg tactic supports any Lean expression.

## 13.1 lean-egg

### 13.1.1 Methodology

The basic methodology of equality saturation is based on E-Graphs, a data structure that can store equivalence classes of terms efficiently. We used the lean-egg tactic (https://github.com/marcusrossel/lean-egg), based on equality saturation as a tactic, which (re)constructs a proof from the E-graph [4] in Lean. This means that we do not have to trust either the egg tool nor the tactic: if something goes wrong, Lean will not accept the constructed proof. In fact, we found issues with the proof reconstruction from the examples in this project.

The lean-egg tactic works for equational reasoning, i.e. proving equalities as consequences of other equalities (potentially universally quantified), which is exactly what we need to prove implications of laws in Magmas. In many cases, we have laws of the form $x = y$, where neither set of variables in the left- and right-hand-side of the law is a subset of each other. In this case the laws cannot be used as rewrite rules: it's not clear what it would be rewritten to, since there are unknowns on both sides of the equation. For these cases we used a simple heuristic, where we instantiate the variables with terms found in the (proof) context, as those are likely to be important for proving the equality.

### 13.1.2 Results

Out of the possible implications between the 34 equations considered in Chapter 2, this method found an additional 86 implications that were not found before. Some of these seem to be missing in the computation of the transitive closure of implications of the equalities (an investigation is in progress), but some of these are genuinely new theorems, and the `lean-egg` tactic finds good proofs of these (these can be rewritten using `calc` style with a different tactic, `calcify`: https://github.com/nomeata/lean-calcify). An example of this is the following proof, found by lean-egg:

**Theorem 13.1** (14 implies 23)**.** *Definition 2.9 is equivalent to Definition 2.11.*

*Proof.*

$$x = (x \diamond x) \diamond (x \diamond (x \diamond x)) = (x \diamond x) \diamond x$$

□

It was also able to (re)prove Theorem 5.5, albeit with a manually-provided hint (guide, in the sense of [4]).

## 13.2 MagmaEgg

This is a simple but apparently at least somewhat effective Rust theorem prover based on egg e-graph library written for this project.

It proved 5574 of the 24283 implications in the "only_strongest.txt" file at the time.

The code was originally based on the magma_search pull request, but has been pretty much completely rewritten.

Currently search just uses the egg library in a basic fashion, except that in case there are extra variables not present in the LHS, it has code to instantiate them with all subexpressions of the original goal.

Exporting the proofs to Lean has turned out to be harder than finding the proofs, but a good solution has been implemented (modulo some issues in egg that require to sometimes turn off explanation optimization since it sometimes triggers stack overflows and assert failures) that directly produces proof terms using let/have and Eq.refl, Eq.symm, Eq.trans, Magma.op, a congruence lemma for Magma.op and variables and the hypothesis. I define one letter aliases for them to reduce verbosity.

Possible future work:

- Figure out which implications are important to prove and try it on them
- Replace the fork-based code with self-execution so that it works on Windows and is less of na hack
- Fork egg and fix the buggy and slow length optimization of explanations
- Maybe write Lean code directly instead of writing explanation sexps and converting to Lean code in a second run
- Fix the generation of extra variable values so it doesn't take too much time in pathological cases (i.e. goals with 4-6 variables)
- Determine whether it actually has some advantages compared to Vampire and lean-egg
- Support searching for multiple goal equations at once
- Write a custom elaborator for Lean to speed up elaboration
- If the Lean kernel turns out to be too slow for some large necessary proofs and thus the custom elaborator is not enough, write a custom verified typechecker
- Support having extra rewrite rules, such as other implications that have been found implied by the hypothesis, or simple equalities found by the egraph search itself
- Run it with massive computing resources if deemed useful and someone offers those, once it's a bit more mature

# Bibliography


[1] A. K. Austin. A note on models of identities. *Proc. Amer. Math. Soc.*, 16:522–523, 1965.

[2] A. Kisielewicz. Austin identities. *Algebra Universalis*, 38(3):324–328, 1997.

[3] Andrzej Kisielewicz. Varieties of algebras with no nontrivial finite members. In *Lattices, semigroups, and universal algebra (Lisbon, 1988)*, pages 129–136. Plenum, New York, 1990.

[4] Thomas Koehler, Andrés Goens, Siddharth Bhat, Tobias Grosser, Phil Trinder, and Michel Steuwer. Guided equality saturation. *Proc. ACM Program. Lang.*, 8(POPL):1727–1758, 2024.

[5] N. S. Mendelsohn and R. Padmanabhan. Minimal identities for Boolean groups. *J. Algebra*, 34:451–457, 1975.

[6] Max Willsey, Chandrakana Nandi, Yisu Remy Wang, Oliver Flatt, Zachary Tatlock, and Pavel Panchekha. egg: Fast and extensible equality saturation. *Proc. ACM Program. Lang.*, 5(POPL):1–29, 2021.