

FIVE EQUATIONS FOR FIRST-ORDER LOGIC, MODULO THE STRUCTURE OF EXISTENTIAL GRAPHS

ANTHONY HART

ABSTRACT. Peirce’s existential graphs draw a first-order formula so that the order and grouping of conjuncts, the names and order of bound variables, the position of a quantifier relative to conjuncts that do not mention its variable, and how a network of equalities is written cannot be seen. We take these structural laws for granted. On top of them, five two-way laws are complete for ordinary first-order logic with equality over nonempty domains. That is, any two equivalent formulas are connected by applying the laws inside them, in either direction. The laws are double negation, deiteration $X \wedge \neg(X \wedge Y) \Leftrightarrow X \wedge \neg Y$, annihilation $\perp \wedge X \Leftrightarrow \perp$, substitution of equals, and existential introduction $X(a) \wedge \exists u.X(u) \Leftrightarrow X(a)$, and none of them follows from the other four. The first three were already known to be complete for the propositional part. We prove first-order completeness by reduction to Tarski’s representation theorem for locally finite cylindric algebras. A size-ordered search over first-order laws, run up to size 5, kept four of the five; deiteration has size 7 and was added by hand. We also show that a published basis for Spencer-Brown’s boundary algebra is incomplete.

1. INTRODUCTION

Peirce’s existential graphs draw a first-order formula on a sheet. Conjunction is drawing side by side, negation is drawing an enclosure around a subgraph, and a quantified variable is drawn as a line (Peirce’s *line of identity*), with lines joined to express equality. Much of the syntax of a formula is therefore not visible in the drawing: the order and grouping of conjuncts, the names and order of bound variables, which conjunct a quantifier is attached to when the others do not mention its variable, and which of several equivalent sets of equalities joins the same variables. We take for granted the laws that identify formulas differing only in these respects, the *structural laws* (Definition 2), and ask which two-way laws must be added to obtain all of first-order logic.

Five suffice: double negation, deiteration $X \wedge \neg(X \wedge Y) \Leftrightarrow X \wedge \neg Y$, annihilation $\perp \wedge X \Leftrightarrow \perp$, substitution of equals, and existential introduction (Definition 5). Together with the structural laws, they connect any two formulas that are equivalent in ordinary first-order logic with equality over nonempty domains, by rewriting subformulas in either direction. None of the five follows from the other four.

Our motivation is a constraint-handling language whose store is a formula held as an existential graph, and whose programs may invoke logical equivalences. Such a system gets the structural laws from its representation for free, and the laws a programmer uses on top should be few and recognisable.

For the propositional part such a set is known. Spencer-Brown reads juxtaposition as disjunction; under the dual reading, his reflexion, generation and integration [25] are double negation, deiteration and annihilation, and Kauffman and Collings adopt them, following Bricken, as a complete basis [15, 7].

Date: September 29, 2026.

For first-order logic, two complete presentations are available, and neither suits rewriting. The cylindric-algebra axioms of Henkin, Monk and Tarski (Section 4.2) are two-way equations, but they need a separate Boolean basis such as Huntington's equation, and their only substitution axiom, (C7), says that $x(v_i)$ and $\neg x(v_j)$ cannot both hold when $v_i = v_j$: it rewrites to false, never $x(v_i)$ to $x(v_j)$. The calculus of neo-Peircean relations of Bonchi, Di Giorgio, Haydon and Sobociński [2, 3] is complete, but its proofs are chains of inequalities, it allows the empty domain, and it has 64 axioms.

Our contributions are:

- (1) *Five laws that, with the structural laws, are complete for first-order logic with equality over nonempty domains* (Theorem 9), and independent when the signature has a relation symbol of positive arity (Theorem 15).
- (2) *A direct proof for the propositional part* (Theorem 7): a four-step derivation of Huntington's equation from double negation, deiteration and annihilation, and the independence of the three; neither appears in the works cited in Section 6. Meguire's published basis [20] for boundary algebra (Spencer-Brown's calculus of nested enclosures, equivalent to Boolean algebra), which uses $\neg\neg\top \Leftrightarrow \top$ in place of double negation, is incomplete (Proposition 8).
- (3) *A search for recognisable laws* (Section 5). Enumerating valid equations by size and keeping those not derived from smaller ones yields, among others, double negation, idempotence, annihilation, non-contradiction, absorption, substitution of equals and existential introduction. Neither search keeps deiteration.

Section 2 fixes the formulas, the structural laws and the five laws. Section 3 treats the propositional part, whose derivations Section 4 reuses; Section 4 proves completeness and independence. Section 5 describes the search, Section 6 compares with prior work, and Appendix A gives the derivations behind the search.

2. FORMULAS, STRUCTURAL LAWS AND THE FIVE LAWS

Definition 1 (Formulas). Fix a relational signature Σ and an infinite set of variables. Formulas are given by

$$\varphi, \psi ::= \top \mid R(w_1, \dots, w_n) \mid w = w' \mid \varphi \wedge \psi \mid \neg\varphi \mid \exists w. \varphi$$

where $R \in \Sigma$ has arity n . A formula $w = w'$ is an *equality*. We write $\perp$ for $\neg\top$ and $\text{fv}(\varphi)$ for the free variables of φ . Disjunction, implication and $\forall$ are the usual abbreviations.

Definition 2 (Structural identity). *Structural identity* $=_s$ is the least congruence on formulas generated by the following *structural laws*:

- (S1) $\wedge$ is associative and commutative with unit $\top$;
- (S2) bound variables may be renamed;
- (S3) $\exists w.(\varphi \wedge \psi) =_s (\exists w.\varphi) \wedge \psi$ when $w \notin \text{fv}(\psi)$, and $\exists w.\exists v.\varphi =_s \exists v.\exists w.\varphi$;
- (S4) $w = w =_s \top$, $w = v =_s v = w$, $w = v \wedge v = z =_s w = v \wedge w = z$, and $w = v \wedge w = v =_s w = v$.

Up to $=_s$, a formula is a tree. Each node is the top level or a negation, and holds a multiset of atoms and negated subformulas, the variables quantified at that node, and the partition of variables that its equalities generate. Existential-graph syntax gives the structural laws for free: formulas related by $=_s$ are drawn as the same graph, so a system that works on graphs never has to apply (S1)–(S4).

Each structural law is invisible in a graph for a concrete reason. (S1): conjuncts drawn side by side on a sheet have no order or grouping. (S2): a line is merely drawn, with no name to distinguish it from another line, so renaming a bound variable changes nothing in the graph.

(S3): a quantified variable is a line, and a line has no position relative to a subgraph it does not touch, so the two can be moved past each other freely; nor do two lines have an order. (S4): equalities are drawn by joining lines, so $w=v \wedge v=z$ and $w=v \wedge w=z$ draw the same three joined lines.

The structural laws are valid equivalences of first-order logic, so they act as further axioms: “five laws” means five in addition to (S1)–(S4). Idempotence of conjunction and vacuous quantification are not structural; they follow from the five laws (Section 4.1).

Definition 3 (Meaning). A structure M has a *nonempty* universe U and a relation $R^M \subseteq U^n$ for each $R \in \Sigma$. Satisfaction $M, s \models \varphi$ under an assignment s of elements to variables is the usual one. Formulas φ and ψ are *equivalent*, $\varphi \equiv \psi$, when $M, s \models \varphi \iff M, s \models \psi$ for every M and s .

Definition 4 (Laws and instances). A *law* is an equation $L \Leftrightarrow R$ between *schemas*: formulas that may also contain *schema variables* X, Y and *schema parameters* a, b , which stand for object variables. A schema variable may take arguments, as in $X(a)$; each schema variable in the five laws takes at most one. An *instance* replaces each parameter by a variable and each schema variable by a formula: a unary X becomes an abstraction $\lambda x.\varphi$, and $X(w)$ becomes the capture-avoiding substitution $\varphi[w/x]$, renaming bound variables of φ as (S2) allows. The formula φ may have free variables besides x , and the law’s own bound variables, such as u in EI below, are renamed to differ from all of them.

So a schema variable stands for any formula, including one that mentions variables the law does not name. This is the usual reading of a schema variable: a law valid for every predicate is valid for every predicate with extra fixed parameters.

Definition 5 (The five laws).

(DN) double negation	$\neg\neg X \Leftrightarrow X$
(DT) deiteration	$X \wedge \neg(X \wedge Y) \Leftrightarrow X \wedge \neg Y$
(AN) annihilation	$\perp \wedge X \Leftrightarrow \perp$
(SUB) substitution of equals	$a=b \wedge X(a) \Leftrightarrow a=b \wedge X(b)$
(EI) existential introduction	$X(a) \wedge \exists u.X(u) \Leftrightarrow X(a)$

Derivability $\sim$ is the least congruence on formulas (for $\wedge$, $\neg$ and $\exists w$) that contains $=_s$ and every instance of the five laws.

Three relations on formulas are now in play: $\varphi =_s \psi$, the same up to the structural laws; $\varphi \sim \psi$, connected by the five laws; and $\varphi \equiv \psi$, logically equivalent. Theorem 9 says $\sim$ and $\equiv$ coincide.

Because $\sim$ is a congruence, a law applies to any subformula, to any part of a conjunction, and in both directions. In particular it may rewrite a subformula in which w is free although w is bound further out.

DT is the simplest case of Peirce’s deiteration: when X is a conjunct beside $\neg(X \wedge Y)$, the copy of X inside may be erased; Peirce also erases copies nested deeper. AN is Spencer-Brown’s integration, and is also called domination. EI is the equational form of the rule $\varphi(a) \rightarrow \exists u.\varphi(u)$: a formula absorbs a copy of itself in which some occurrences of a variable are replaced by a fresh bound one.

Remark 6 (Where nonempty domains enter). Instantiating EI with $X := \lambda x.\top$ gives the law of vacuous quantification,

$$(U) \quad \exists u.\top \sim \top.$$

The instance must supply a variable for the parameter a , which neither side mentions; Definition 4 allows that, and nonemptiness is used exactly there.

An implementation may instead work in context: every free variable of a law instance, including those of the formulas substituted for schema variables, must be bound above the point of application or free in the formula being rewritten as a whole. Then $\cup$ is not derivable between sentences, and must be added as a sixth law. To see this, note that a derivation between sentences then passes only through sentences. Evaluate them over the empty domain, where every $\exists w.\chi$ is false. A rewrite below a quantifier cannot change the truth value. At the top level no variable is in scope, so SUB and EI cannot apply there, and DN, DT, AN and (S1)–(S4) are valid. So every equivalence between sentences derivable in context holds over the empty domain, but $\exists u.\top \Leftrightarrow \top$ does not.

3. THE PROPOSITIONAL PART

We treat the propositional part first because Section 4 reuses its derivations. In this section formulas are Boolean terms: x, y range over propositions, $x \cdot y$ is $x \wedge y$, and $-x$ is $\neg x$. Let $\perp := -\top$.

Theorem 7. *The equations*

$$\begin{aligned} x \cdot y &= y \cdot x, & (x \cdot y) \cdot z &= x \cdot (y \cdot z), & x \cdot \top &= x, \\ (\text{DN}) \quad - -x &= x, & (\text{DT}) \quad x \cdot -(x \cdot y) &= x \cdot -y, & (\text{AN}) \quad \perp \cdot x &= \perp \end{aligned}$$

axiomatise Boolean algebras, with $\cdot$ as meet, $-$ as complement and $\top$ as top. Each of DN, DT, AN is independent of the other five equations in this list.

The three laws are Spencer-Brown's Consequences 1–3, dualised, which Kauffman and Collings take as a complete basis (Section 6). We give a direct proof.

Proof. All six equations hold in every Boolean algebra. For the converse we derive Huntington's equation, written with $\cdot$ in place of his $+$:

$$(\text{H}) \quad -(-x \cdot y) \cdot -(-x \cdot -y) = x.$$

Huntington proved that commutativity, associativity and (H), written with $+$, axiomatise Boolean algebras with $+$ as the join [13, 17]. So our equations make $\cdot$ the join of some Boolean order. Under the reverse order $\cdot$ is the meet, and $-$ is still the complement, because complement is an order-reversing involution. The unit law then says $\top$ is the top. The derivation has four steps.

- (1) $-\perp = \top$: $-\perp = - -\top = \top$ by DN.
- (2) *Contradiction*, $x \cdot -x = \perp$: $x \cdot -x = x \cdot -(x \cdot \top) = x \cdot -\top = \perp \cdot x = \perp$, by the unit, DT with $y := \top$, commutativity and AN.
- (3) *Absorption*, $x \cdot -(-x \cdot y) = x$: $x \cdot -(-x \cdot y) = x \cdot -(x \cdot -x \cdot y) = x \cdot -(\perp \cdot y) = x \cdot -\perp = x \cdot \top = x$, by DT read right to left, contradiction (2), AN, step 1 and the unit.
- (4) (H): put $B := -(-x \cdot y)$. DT with $-x$ in place of its x gives $-x \cdot -(-x \cdot y) = -x \cdot -y$, so $-x \cdot -y = -x \cdot B$, and the second factor of (H) is $-(-x \cdot B)$. The left side of (H) is therefore $B \cdot -(B \cdot -x) = B \cdot - -x = B \cdot x = x \cdot -(x \cdot y) = x$, by commutativity, DT, DN, commutativity and absorption (3).

For independence, each of the following algebras, with $\cdot$ as min and $\top$ as the largest element, satisfies commutativity, associativity, the unit and two of the three laws, and falsifies the third:

falsified	universe	complement	failing instance
DN	$\{0, 1\}$	$-x = 0$	$--1 = 0$
DT	$\{0, 1, 2\}$	$-0 = 2, -1 = 1, -2 = 0$	$1 \cdot -(1 \cdot 2) = 1 \neq 0 = 1 \cdot -2$
AN	$\{0, 1\}$	$-x = x$	$\perp \cdot 0 = 0 \neq 1 = \perp$

The second is Kleene's strong three-valued logic. A computer check over all argument values confirms each row. $\square$

Proposition 8. *Meguire's basis for boundary algebra [20, §3] does not axiomatise Boolean algebras. In our notation it consists of commutativity and associativity (his B1), $--\top = \top$ (B2), $-x \cdot x = \perp$ (B3) and DT (B4).*

Proof. Take the universe $\{e, p, q\}$ with $\top = e$, the product given by $e \cdot x = x$, $p \cdot p = e$ and $q \cdot x = q$, and complement $-e = q$, $-p = q$, $-q = e$. All of B1–B4 and the unit hold, but $--p = e \neq p$ and $p \cdot p = e \neq p$. The model was found by the finite-model finder Mace4 [18] and re-checked by a separate computer evaluation. Meguire derives $x \cdot x = x$ and $--x = x$ with a “calculation” rule, which infers $\alpha = \beta$ from $(\alpha \leftrightarrow \beta) = \top$. That rule is not an equational consequence of B1–B4, and it fails in this model. $\square$

4. FIRST-ORDER LOGIC: COMPLETENESS AND INDEPENDENCE

We now prove that the five laws, with the structural laws, connect exactly the logically equivalent formulas, and that none of the five can be dropped.

Theorem 9. *For all formulas φ and ψ , $\varphi \equiv \psi$ if and only if $\varphi \sim \psi$.*

Soundness is routine: each law is valid for every choice of formulas, including formulas with extra free variables, and $=_s$ preserves meaning. Sections 4.1 to 4.3 prove completeness: we translate formulas into cylindric-algebra terms, use Tarski's theorem there, and translate the resulting equational proof back. Section 4.4 proves independence.

4.1. Derived laws. The proof uses these consequences of the five laws; u is always a bound variable not free elsewhere.

(U) vacuous quantification	$\exists u. \top \sim \top$	EI with $X := \lambda x. \top$
(I) a variable has a witness	$\exists u. (u = a) \sim \top$	EI with $X := \lambda x. (x = a)$
(OP) one-point rule	$\exists u. (u = a \wedge X(u)) \sim X(a)$	SUB, (S3), I
(CTR) contradiction	$X \wedge \neg X \sim \perp$	step 2 of Theorem 7
(ABS) absorption	$X \wedge \neg(\neg X \wedge Y) \sim X$	step 3 of Theorem 7

For I, the instance of EI reads $(a=a) \wedge \exists u. (u=a) \Leftrightarrow a=a$, and $a=a =_s \top$ by (S4). For OP: $\exists u. (u=a \wedge X(u)) \sim \exists u. (u=a \wedge X(a)) =_s (\exists u. u=a) \wedge X(a) \sim X(a)$. The derivations in Theorem 7 use only DN, DT, AN and (S1), so they hold with $\wedge$ for $\cdot$, $\neg$ for $-$, and arbitrary formulas for x and y . Idempotence also follows: $\exists u. X =_s X \wedge \exists u. \top \sim X$ by (S3) and U, so $X \wedge X \sim X \wedge \exists u. X$, which is $\sim X$ by EI with its schema variable set to $\lambda x. X$, which ignores x .

4.2. Cylindric algebras. We use the axioms of Henkin, Monk and Tarski as stated by Monk [22]. Because we will generate a congruence from the axioms, we need the Boolean part as equations. We take commutativity, associativity and unit of $\cdot$, Huntington's equation (H), and $0 = -1$, and define $x + y := -(x \cdot -y)$; by the proof of Theorem 7 these axiomatise Boolean algebras with $\cdot$ as meet. A *cylindric algebra of dimension ω* is such a Boolean algebra

$(A, \cdot, -, 1)$ with operations c_i (read as $\exists v_i$) and constants $d_{i,j}$ (read as $v_i = v_j$) for $i, j < \omega$, such that

$$\begin{aligned}
 (C1) \quad & c_i 0 = 0 & (C2) \quad & x + c_i x = c_i x \\
 (C3) \quad & c_i(x \cdot c_i y) = c_i x \cdot c_i y & (C4) \quad & c_i c_j x = c_j c_i x \\
 (C5) \quad & d_{i,i} = 1 & (C6) \quad & d_{j,k} = c_i(d_{j,i} \cdot d_{i,k}) \text{ for } i \notin \{j, k\} \\
 (C7) \quad & c_i(d_{i,j} \cdot x) \cdot c_i(d_{i,j} \cdot -x) = 0 \text{ for } i \neq j.
 \end{aligned}$$

$\mathbf{CA}_\alpha$ is the class of cylindric algebras of dimension α .

The *dimension set* of x is $\{i : c_i x \neq x\}$, and an algebra is *locally finite* if every dimension set is finite. $\mathbf{Lf}_\omega$ is the class of locally finite algebras in $\mathbf{CA}_\omega$.

The *full cylindric set algebra* with base U is $(\mathcal{P}({}^\omega U), \cap, S \mapsto {}^\omega U \setminus S, {}^\omega U, C_i, D_{i,j})$, where ${}^\omega U$ is the set of sequences $s : \omega \rightarrow U$, $C_i S = \{s : s[i \mapsto e] \in S \text{ for some } e \in U\}$ and $D_{i,j} = \{s : s_i = s_j\}$. $\mathbf{Cs}_\omega$ is the class of its subalgebras, for all U . For a class K , $\mathbf{SP}K$ is the class of algebras isomorphic to subalgebras of products of members of K . An algebra is *representable* if it lies in $\mathbf{RCA}_\omega := \mathbf{SP} \mathbf{Cs}_\omega$.

Theorem 10 (Tarski). *Every locally finite cylindric algebra of dimension ω is representable: $\mathbf{Lf}_\omega \subseteq \mathbf{RCA}_\omega$.*

Monk writes: “That every infinite dimensional locally finite $\mathbf{CA}_\alpha$ is representable is due to Tarski” [22]. Andr eka, N emeti and Sain use $\mathbf{Lf}_\omega \subseteq \mathbf{RCA}_\omega$ in the proof of their Theorem 7.9 and give $\mathbf{RCA}_\omega = \mathbf{SP} \mathbf{Cs}_\omega$ as their Theorem 2.4(iii) [1], citing the proofs in [12].

4.3. Translations. Only finitely many relation symbols occur in φ and ψ , and equivalence over Σ -structures coincides with equivalence over structures for those symbols, so assume Σ is finite. Let N bound its arities. Let $\mathbf{F}$ be the algebra of terms built from one constant R for each $R \in \Sigma$ with the cylindric operations. Let $\approx$ be the congruence on $\mathbf{F}$ generated by every instance of the cylindric axioms (any terms for the axioms’ variables) together with the *dimension equations* $c_i R = R$ for $i \geq \text{ar}(R)$. So $\approx$ is to terms what $\sim$ is to formulas.

A term t has a meaning $\llbracket t \rrbracket_M \subseteq {}^\omega U$ in each structure M : the operations act as in the full set algebra, and $\llbracket R \rrbracket_M = \{s : (s_0, \dots, s_{n-1}) \in R^M\}$. In other words a term is a first-order formula over the variables $0, 1, 2, \dots$ in which an atom R always has the variables $0, \dots, n-1$ as arguments; c_i is $\exists$ over variable i , and $d_{i,j}$ says that variables i and j are equal. This is the ranked first-order logic of [1, §7].

By (S2), rename the bound variables of φ and ψ so that no variable is bound twice and none is both bound and free. Choose an injection κ from their variables into $\{N, N+1, \dots\}$; coordinates $0, \dots, N-1$ are then left for the arguments of atoms. The translation tr sends

$$\begin{aligned}
 \top &\mapsto 1, \quad \chi \wedge \theta \mapsto \text{tr } \chi \cdot \text{tr } \theta, \quad \neg \chi \mapsto -\text{tr } \chi, \quad \exists w. \chi \mapsto c_{\kappa w} \text{tr } \chi, \quad (w = v) \mapsto d_{\kappa w, \kappa v}, \\
 R(w_1, \dots, w_n) &\mapsto c_0 \cdots c_{n-1} (d_{0, \kappa w_1} \cdots d_{n-1, \kappa w_n} \cdot R).
 \end{aligned}$$

The atom case sets coordinate $m-1$ equal to coordinate κw_m for each m , applies R to coordinates $0, \dots, n-1$, and quantifies those coordinates away. Since every $\kappa w \geq N \geq n$, induction on χ gives

$$(1) \quad s \in \llbracket \text{tr } \chi \rrbracket_M \iff M, s \circ \kappa \models \chi,$$

where $s \circ \kappa$ is the assignment $w \mapsto s_{\kappa w}$.

Conversely, fix distinct variables $v_0, v_1, \dots$ with $v_{\kappa w} = w$ for each variable w of φ and ψ , and with every other v_i occurring in neither. Define $\iota : \mathbf{F} \rightarrow$ formulas by $\iota(R) = R(v_0, \dots, v_{n-1})$, $\iota(1) = \top$, $\iota(0) = \perp$, $\iota(t \cdot t') = \iota t \wedge \iota t'$, $\iota(-t) = \neg \iota t$, $\iota(c_i t) = \exists v_i. \iota t$ and $\iota(d_{i,j}) = (v_i = v_j)$.

Lemma 11. *If $t \approx t'$ then $\iota t \sim \iota t'$.*

Proof. The relation $\{(t, t') : \iota t \sim \iota t'\}$ is a congruence on $\mathbf{F}$, because ι is defined compositionally and $\sim$ is a congruence. So it suffices that ι sends each generating equation to a derivable one. Write $P = \iota x$ and $Q = \iota y$.

- Commutativity, associativity and the unit hold by (S1). $0 = -1$ holds by definition of ι . Huntington's (H) was derived from DN, DT, AN in Theorem 7.
- (C1): $\exists v_i. \perp =_s \perp \wedge \exists v_i. \top \sim \perp$ by (S3) and AN.
- (C2): write $E := \exists v_i. P$. By EI with $X := \lambda v_i. P$ and $a := v_i$, $P \sim P \wedge E$. So $\iota(x + c_i x) = \neg(\neg P \wedge \neg E) \sim \neg(\neg(P \wedge E) \wedge \neg E)$. By absorption (ABS, with $\neg E$ for X and P for Y) and DN, $\neg E \wedge \neg(E \wedge P) \sim \neg E$, so $\iota(x + c_i x) \sim \neg \neg E \sim E = \iota(c_i x)$.
- (C3): $\exists v_i. (P \wedge \exists v_i. Q) =_s (\exists v_i. P) \wedge (\exists v_i. Q)$ by (S3), since $v_i \notin \text{fv}(\exists v_i. Q)$. (C4) is the second clause of (S3).
- (C5): $v_i = v_i =_s \top$ by (S4).
- (C6): $\exists v_i. (v_j = v_i \wedge v_i = v_k) =_s (\exists v_i. v_i = v_j) \wedge v_j = v_k \sim v_j = v_k$ by (S4), (S3) and $\top$ ($\exists u. u = a \sim \top$). This covers $j = k$, where $v_j = v_k =_s \top$.
- (C7): apply the one-point rule OP with $u := v_i$ and $a := v_j$, once with $X := \lambda v_i. P$ and once with $X := \lambda v_i. \neg P$. This gives $\exists v_i. (v_i = v_j \wedge P) \wedge \exists v_i. (v_i = v_j \wedge \neg P) \sim P[v_j/v_i] \wedge \neg P[v_j/v_i]$, which is $\sim \perp$ by contradiction (CTR).
- Dimension equations: $\exists v_i. R(v_0, \dots, v_{n-1}) =_s R(v_0, \dots, v_{n-1}) \wedge \exists v_i. \top \sim R(v_0, \dots, v_{n-1})$ by (S3) and vacuous quantification (U), since $i \geq n$. $\square$

Lemma 12. $\chi \sim \iota(\text{tr } \chi)$ for every formula χ whose variables are in the domain of κ .

Proof. By induction on χ ; every case but atoms is immediate from the definitions and congruence. For an atom,

$$\iota(\text{tr } R(w_1, \dots, w_n)) = \exists v_0 \dots \exists v_{n-1}. (v_0 = w_1 \wedge \dots \wedge v_{n-1} = w_n \wedge R(v_0, \dots, v_{n-1})),$$

where the v_m differ from the w 's because $\kappa w \geq N$. Using (S3) to move each quantifier next to its equality, n applications of the one-point rule OP give $R(w_1, \dots, w_n)$. $\square$

Lemma 13. *If $\llbracket t \rrbracket_M = \llbracket t' \rrbracket_M$ for every structure M , then $t \approx t'$.*

Proof. The quotient $\mathbf{F}/\approx$ satisfies the cylindric axioms, since every element is the class of a term and all instances were identified. It is locally finite: $c_k[t] = [t]$ whenever $k \geq N$ and k does not occur in t , so the dimension set of $[t]$ lies in $\{0, \dots, N-1\}$ together with the indices occurring in t , a finite set. This follows by induction on t from the dimension equations and these standard consequences of the axioms, the last three under the hypothesis $c_k x = x$ and $c_k y = y$:

- $c_k 0 = 0$ (C1), $c_k 1 = 1$ (C2 with $x := 1$) and $c_k c_k y = c_k y$ (C3 with $x := 1$);
- $c_k d_{i,j} = d_{i,j}$ for $k \notin \{i, j\}$, from (C6) and $c_k c_k = c_k$;
- $c_k(x \cdot y) = c_k(x \cdot c_k y) = c_k x \cdot c_k y = x \cdot y$ (C3);
- $c_k(-x) = -x$: by (C3) and (C1), $c_k(-c_k x) \cdot c_k x = c_k(-c_k x \cdot c_k x) = c_k 0 = 0$, so $c_k(-c_k x) \leq -c_k x$; with (C2) this is equality;
- $c_k c_i x = c_i c_k x = c_i x$ by (C4).

Suppose $t \not\approx t'$. By Theorem 10 there is a homomorphism h from $\mathbf{F}$ into a full cylindric set algebra over some base U with $h(t) \neq h(t')$: compose the quotient map, the embedding into a product of set algebras, and a projection that separates the two images. Then $U \neq \emptyset$, since over the empty base the algebra has one element. Pick $z \in h(t) \triangle h(t')$, and let W be the set of sequences that differ from z in finitely many places. Any two sequences in W differ in finitely many coordinates, and the dimension equations let us change one coordinate at a time,

so finitely many such steps connect them. Changing one coordinate of a sequence in W keeps it in W , so $S \mapsto S \cap W$ preserves the Boolean operations (relative to W) and C_i , and sends $D_{i,j}$ to $D_{i,j} \cap W$.

Let M have universe U , and for R of arity n let $R^M = \{(s_0, \dots, s_{n-1}) : s \in h(R) \cap W\}$. Then $h(R) \cap W = \llbracket R \rrbracket_M \cap W$. Indeed, if $s \in W$ agrees below n with some $s' \in h(R) \cap W$, then s and s' differ in finitely many coordinates, all at least n . Since h preserves the dimension equations, $C_i h(R) = h(R)$ for $i \geq n$, so changing those coordinates one at a time keeps us in $h(R)$. By induction on terms, $h(r) \cap W = \llbracket r \rrbracket_M \cap W$ for every term r . As $z \in W$, z lies in exactly one of $\llbracket t \rrbracket_M$ and $\llbracket t' \rrbracket_M$, contradicting the hypothesis. $\square$

Proof of Theorem 9. Let $\varphi \equiv \psi$, renamed and translated as above. By (1), $\llbracket \text{tr } \varphi \rrbracket_M = \llbracket \text{tr } \psi \rrbracket_M$ for every M . So $\text{tr } \varphi \approx \text{tr } \psi$ by Lemma 13, and $\iota \text{tr } \varphi \sim \iota \text{tr } \psi$ by Lemma 11. By Lemma 12, $\varphi \sim \psi$. $\square$

Remark 14 (Concrete formulas, not schemas). Theorem 9 is about concrete formulas: every instance of a valid law is derivable. It does not say that a valid law is derivable *uniformly*, by one derivation in which its schema variables stay opaque symbols. We do not claim that. Monk's theorem suggests it fails: no finite set of equations in the cylindric signature axiomatises the representable cylindric algebras [21]. We have not proved the translation from uniform derivability to cylindric equations that would make this a consequence.

4.4. Independence.

Theorem 15. *Suppose Σ contains a relation symbol R of positive arity. Then none of the five laws is derivable from the other four together with $=_s$.*

The hypothesis is used for DT, AN and SUB, whose counterexamples below need an atom; for $\Sigma = \emptyset$ we have not settled their independence.

Proof. For each law we give an interpretation, compositional in the formula constructors and invariant under $=_s$, that validates the other four laws for all instances and falsifies one instance of the law in question.

- DN, DT, AN: forget variables. Use the three algebras with $\cdot = \min$ tabulated in the proof of Theorem 7. Send $\wedge$ to $\cdot$, $\neg$ to $-$, $\top$ and every equality to $\top$, and $\exists w.\chi$ to the value of χ ; assign each relation symbol a value, which every atom with that symbol takes, whatever its arguments. SUB becomes $\top \cdot x = \top \cdot x$, and EI becomes $x \cdot x = x$, which holds because $\min$ is idempotent. (S1)–(S4) hold trivially. DN fails at $\neg\neg\top$ in the first algebra. With R given the value 1, the instance $R \wedge \neg(R \wedge \top) \Leftrightarrow R \wedge \neg\top$ of DT takes values 1 and 0 in the second; with R given the value 0, the instance $\perp \wedge R \Leftrightarrow \perp$ of AN takes values 0 and 1 in the third (writing R for an atom $R(w, \dots, w)$).
- SUB: interpret $=$ as an arbitrary equivalence relation on U rather than equality. (S4) holds because the relation is an equivalence, and DN, DT, AN, EI do not depend on what $=$ means. With $U = \{0, 1\}$, $=$ total and $R^M = \{0\}$, the instance $a=b \wedge R(a) \Leftrightarrow a=b \wedge R(b)$ fails at $a \mapsto 0, b \mapsto 1$.
- EI: interpret every $\exists w.\chi$ as false. (S3) holds because both sides of each of its equations are then false; SUB still holds because quantified subformulas are constant; DN, DT, AN are Boolean identities. The instance $\exists u.\top \Leftrightarrow \top$ fails. $\square$

5. SEARCHING FOR THE LAWS

The five laws were chosen to be few and recognisable (Section 1). To find recognisable candidates we searched valid equations in order of size. The first-order search, run up to size

5, kept four of the five laws; deiteration has size 7 and was added by hand. The method follows Wolfram’s listing of Boolean theorems by size. He observes that “with the exception of the second distributive law” the textbook’s named theorems “are exactly the ones that cannot be derived from preceding theorems in the list” [26, p. 817]. QuickSpec applies the same filter to discover laws of programs [24]. The method is:

- (1) enumerate valid equations in order of total size, up to the structural laws;
- (2) try to derive each from the equations kept so far;
- (3) keep it if that fails.

Minimality is not the goal, so step 2 may be a bounded search: its only possible error is keeping a redundant law.

5.1. Propositional search. The propositional search tests the method against a known complete basis. Equations over at most four variables in $(\cdot, -, \top)$ were listed up to associativity, commutativity and unit, and ordered by size, the number of symbol occurrences. Each was given to Prover9 [18] for 3 seconds with the equations kept so far as hypotheses. If Prover9 failed, the equation was kept, and Mace4 searched for a model of the earlier kept equations in which the new one fails. Every kept equation received such a countermodel, so none is derivable from the ones before it. After each addition, Prover9 had 30 seconds to derive Huntington’s equation (H).

Up to size 10 there are 571 valid equations. Seven were kept:

equation	elements in countermodel	name
1 $--\top = \top$	2	double negation of $\top$
2 $--x = x$	2	double negation
3 $x \cdot x = x$	2	idempotence
4 $\perp \cdot x = \perp$	2	annihilation
5 $-(-x \cdot x) = \top$	3	non-contradiction
6 $-(-x \cdot y) \cdot x = x$	6	absorption
7 $-(-(x \cdot y) \cdot x) \cdot y = y$	6	(no standard name)

After entry 7, Prover9 derived (H) in 0.04 seconds. Deiteration also has size 10, but the order within a size placed it after entry 7, and Prover9 derived it from the seven kept entries, so it was not kept. Prover9 derives (H) from $--x = x$ and entry 7 alone, so these two form a complete basis. We prefer double negation, deiteration and annihilation, because entry 7 is not a move a person would recognise.

A second sweep took eight candidate laws and determined, for each set of at most three of them, whether it derives (H): double negation; its instance $--\top = \top$; idempotence; deiteration under one negation (DT) and under two, $x \cdot -(y \cdot -(x \cdot z)) = x \cdot -(y \cdot z)$; annihilation; contradiction $x \cdot -x = \perp$; and $x \cdot -(\perp \cdot y) = x$. Of the 92 sets, exactly four are complete, each of size three:

- double negation, deiteration and annihilation;
- the same with annihilation replaced by contradiction, or by $x \cdot -(\perp \cdot y) = x$;
- double negation, deiteration under two negations, and contradiction.

The other 88 have countermodels, with 2 elements (57 of them), 3 (30) or 6 (1). Replacing double negation by its instance $--\top = \top$ always loses completeness.

Published bases were also checked, each by asking Prover9 to derive (H) from it: Huntington’s; Robbins’ (with the unit law $x \cdot \top = x$ added); the single axiom DN1 and Meredith’s pair, both as given by McCune et al. [19]; Kauffman’s pair [14]; Spencer-Brown’s single equation,

his Consequence 6 [25], as formalised in [23]; Bricken’s rules [9]; and Meguire’s B1–B4 [20]. All are complete except Meguire’s (Proposition 8).

Bricken’s rules without his involution law ($- - x = x$) do not derive involution: in the two-element model with $\cdot = \min$, $\top = 1$ and $-x = 1$, his other rules hold, including deiteration into every context, while double negation fails.

All 121 countermodels were re-checked by a separate script that evaluates the equations in each model directly: 88 from the subset sweep, 26 from runs on single candidate laws and on the published bases, and 7 from the size-ordered search.

5.2. First-order search. First-order laws were enumerated as equations between schemas. A schema has at most two schema variables, with at most two arguments each, and two schema parameters a, b ; it is built from these with negations, quantifiers and equalities. Size counts schema-variable occurrences, negations, quantifiers and equalities. The equalities at one node count as the number of joins needed to connect their variables: $a=b$ counts 1, and $a=b \wedge b=c$ counts 2. Schemas were enumerated up to structural identity. They were grouped by their truth values in all interpretations over domains of size 1 and 2, and in sampled interpretations over sizes 3 and 4. Prover9 then split each group into equivalence classes, reading a schema variable as an uninterpreted predicate on its arguments; each pair within a class is a valid equation. All 180,301 Prover9 runs succeeded, so no pair was left undecided.

Derivability was tested by a rewriting engine written for the purpose. It represents a formula up to structural identity, as the tree described after Definition 2. A law’s negations match whole negated subformulas, its quantified variables match quantified variables used nowhere outside the match, and its schema variables match parts of a conjunction. The engine runs a bidirectional breadth-first search over formulas in two stages. The first goes to depth 6, visits at most 4,000 formulas, and allows intermediate formulas at most 2 larger than the larger side of the equation. The second goes to depth 7, visits at most 30,000, and allows 4. An equation is kept only if both searches fail, so for first-order laws “kept” means only “not derived within these bounds”. The engine’s soundness was tested on 1,067,173 rewrite steps, each checked to preserve truth values in sampled finite structures.

The outcome depended on how schema variables match. At first a schema variable could only match formulas whose free variables were among its arguments. Under that convention the search kept one variant of non-contradiction for each number of extra free variables, up to the size bound, because a two-argument schema variable cannot match a formula with three free variables. Allowing the matched formula to mention further variables in scope, as in Definition 4, removed these variants.

All 2,417 valid equations between schemas of total size (both sides) at most 5 were each either derived or kept: 1,526 derived directly by the engine, 880 by composing derivations already found, and 11 kept.

size	kept equation	name
1	$\exists u. \top \Leftrightarrow \top$	vacuous quantification (U)
2	$\exists u. a=u \Leftrightarrow \top$	a variable has a witness (I)
2	$\neg \perp \Leftrightarrow \top$	double negation of $\top$
3	$\neg(\perp \wedge X) \Leftrightarrow \top$	(no standard name)
3	$\perp \wedge X \Leftrightarrow \perp$	annihilation
3	$X \wedge X \Leftrightarrow X$	idempotence
4	$a=b \wedge X(a, b) \Leftrightarrow a=b \wedge X(a, a)$	substitution of equals
4	$\neg(\neg X \wedge X) \Leftrightarrow \top$	non-contradiction
4	$\neg \neg X \Leftrightarrow X$	double negation
4	$X(a, b) \wedge \exists u. X(a, u) \Leftrightarrow X(a, b)$	existential introduction
5	$\exists u. \neg(\neg X(a, u) \wedge X(u, b)) \Leftrightarrow \top$	drinker-style law

The kept substitution and existential-introduction equations are SUB and EI of Definition 5 with X also mentioning b , which Definition 4 allows. The drinker-style law says $\exists u (X(u, b) \rightarrow X(a, u))$; it holds by cases, taking $u = a$ or $u = b$, like the drinker principle.

These eleven equations do not derive (H). To see this, forget variables as in the proof of Theorem 15: send equalities to $\top$ and drop quantifiers. The six-element countermodel for entry 7 of the propositional table then satisfies all eleven and falsifies (H). Adding deiteration, which has size 7, suffices. From the five laws alone the engine derived 13 targets, checking every step: U and I; the five kept equations not among the five laws or these two; Huntington's (H); absorption ABS; the one-point rule OP; and the schema forms of (C2), (C6) and (C7) that the completeness proof uses, (C6) being $\exists u. (a=b \wedge a=u) \Leftrightarrow a=b$.

The drinker-style law took 12 steps chosen by hand, with the gaps between them filled by the engine. Appendix A gives every one of these derivations. By Theorem 9, any set of valid laws that derives the five, such as the kept equations together with deiteration, is complete.

6. RELATED WORK

Equational presentations of the propositional part. The three propositional laws come from Spencer-Brown's primary algebra [25], an equational calculus of enclosures with juxtaposition read as disjunction; dualised, his reflexion, generation and integration are DN, DT and AN. Kauffman and Collings adopt these three, credited to a 1986 report of Bricken, "as an axiomatically complete basis for the Primary Algebra" [15, 7]. Bricken presents occlusion, involution and pervasion (deiteration at every depth) as the axioms of boundary logic and relates them to Peirce's alpha graphs [9, 8]. Meguire's boundary algebra [20] is incomplete (Proposition 8).

Peirce's graphs as proof systems. Proof systems for Peirce's graphs use one-way weakening; our laws are all two-way. Brady and Trimble prove that alpha, as an equational theory, is the theory of Boolean algebras [5]. Their equivalence is mutual derivability under Peirce's rules, including weakening, and their completeness proof uses weakening. Their beta calculus is sound and complete for first-order logic, again with weakening, and with rules that cut and splice lines, together generating a preorder [6]. Ma and Pietarinen give a deep-inference reformulation of alpha whose rules include deletion and insertion [16]; Donato's flower calculus is complete for intuitionistic provability [10]. These systems derive theorems; our laws connect any two equivalent formulas without passing through a non-equivalent one.

Diagrammatic first-order logic. Our laws are equations on formulas with negation primitive, not a diagrammatic calculus. The closest diagrammatic calculi either reason by inequalities or derive negation. Haydon and Sobociński give a compositional calculus of beta graphs and conjecture it complete [11]; it uses inequalities and a contraposition rule. Bonchi, Di Giorgio,

Haydon and Sobociński prove a complete axiomatisation of the calculus of neo-Peircean relations [2, 3]. Its proofs are chains of inequalities, negation is derived rather than primitive, empty domains are allowed, and the authors call the axioms “far from minimal”. Bonchi, Di Giorgio and Trotta give an equational presentation of Boolean hyperdoctrines in the same style, again with negation derived [4]. Our result does not settle the Haydon–Sobociński conjecture, whose calculus is different.

Cylindric algebras. The route of Section 4, through Tarski’s representation theorem, is the classical proof of completeness for ranked first-order logic [1, Thm. 7.9]. Our contribution there is the translation showing that the five laws derive every cylindric axiom, given the structural laws.

APPENDIX A. DERIVATIONS OF THE SEARCH TARGETS

Each of the 13 targets of Section 5 follows from the five laws. All but the drinker-style law follow directly from results in Sections 3 and 4:

target	derivation
U: $\exists u. \top \Leftrightarrow \top$	EI with $X := \lambda x. \top$ (Section 4.1)
I: $\exists u. a = u \Leftrightarrow \top$	EI with $X := \lambda x. (x = a)$ (Section 4.1)
$\neg \perp \Leftrightarrow \top$	DN, since $\neg \perp = \neg \neg \top$
$\neg(\perp \wedge X) \Leftrightarrow \top$	AN, then $\neg \perp \Leftrightarrow \top$
$X \wedge X \Leftrightarrow X$	U, (S3), EI (Section 4.1)
$\neg(\neg X \wedge X) \Leftrightarrow \top$	CTR inside the negation, then $\neg \perp \Leftrightarrow \top$
(H)	steps 1–4 of the proof of Theorem 7
ABS: $X \wedge \neg(\neg X \wedge Y) \Leftrightarrow X$	step 3 of the proof of Theorem 7
OP: $\exists u. (u = a \wedge X(u)) \Leftrightarrow X(a)$	SUB, (S3), I (Section 4.1)
(C2): $X(a, b) \wedge \exists u. X(u, b) \Leftrightarrow X(a, b)$	EI at $\lambda x. X(x, b)$
(C6): $\exists u. (a = b \wedge a = u) \Leftrightarrow a = b$	(S4), (S3), I
(C7): $\exists u. (u = a \wedge X(u)) \wedge \exists v. (v = a \wedge \neg X(v)) \Leftrightarrow \perp$	OP twice, then CTR

The remaining target needs one more lemma: from $\neg \exists u. \varphi(u)$ one may add $\neg \varphi(a)$ for any a .

Lemma 16 (Universal instantiation). $\neg \exists u. Z(u) \sim \neg \exists u. Z(u) \wedge \neg Z(a)$.

Proof. Write $S := \exists u. Z(u)$. Absorption ABS, read right to left with $\neg S$ for its X and $Z(a)$ for its Y , gives $\neg S \sim \neg S \wedge \neg(\neg \neg S \wedge Z(a))$. Double negation inside turns this into $\neg S \wedge \neg(S \wedge Z(a))$, and $Z(a) \wedge \exists u. Z(u) \sim Z(a)$ by EI, which leaves $\neg S \wedge \neg Z(a)$. $\square$

Proposition 17 (The drinker-style law, $\exists u (X(u, b) \rightarrow X(a, u))$). $\exists u. \neg(\neg X(a, u) \wedge X(u, b)) \sim \top$.

Proof. Write $D(w) := \neg(\neg X(a, w) \wedge X(w, b))$, which says $X(w, b) \rightarrow X(a, w)$, and $S := \exists u. D(u)$. The idea: if S failed, both $D(b)$ and $D(a)$ would fail, and their failures contradict each other on $X(a, b)$.

$S \sim \neg \neg S$	DN
$\sim \neg(\neg S \wedge \neg D(b) \wedge \neg D(a))$	Lemma 16, twice
$\sim \neg(\neg S \wedge \neg X(a, b) \wedge X(b, b) \wedge \neg X(a, a) \wedge X(a, b))$	DN on $\neg D(b)$ and on $\neg D(a)$
$\sim \neg(\neg S \wedge X(b, b) \wedge \neg X(a, a) \wedge \perp)$	CTR on $X(a, b) \wedge \neg X(a, b)$
$\sim \neg \perp$	AN
$\sim \top$	DN.

Both uses of Lemma 16 are inside the outer negation. The engine’s chain for this derivation has the same shape, with each use of Lemma 16 spelled out as its three steps. $\square$

REFERENCES

- [1] Hajnal Andréka, István Németi, and Ildikó Sain. Algebraic logic. In *Handbook of Philosophical Logic*, volume 2, pages 133–247. Springer, 2nd edition, 2001. Preprint: <https://old.renyi.hu/pub/algebraic-logic/handbook.pdf>.
- [2] Filippo Bonchi, Alessandro Di Giorgio, Nathan Haydon, and Paweł Sobociński. Diagrammatic algebra of first order logic. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2024)*, pages 1–15. ACM, 2024.
- [3] Filippo Bonchi, Alessandro Di Giorgio, Nathan Haydon, and Paweł Sobociński. The calculus of neo-Peircean relations. *Logical Methods in Computer Science*, 22(2):29:1–29:75, 2026.
- [4] Filippo Bonchi, Alessandro Di Giorgio, and Davide Trotta. When Lawvere meets Peirce: An equational presentation of Boolean hyperdoctrines. In *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *LIPIcs*, pages 30:1–30:19. Schloss Dagstuhl, 2024.
- [5] Geraldine Brady and Todd H. Trimble. A categorical interpretation of C.S. Peirce’s propositional logic Alpha. *Journal of Pure and Applied Algebra*, 149(3):213–239, 2000.
- [6] Geraldine Brady and Todd H. Trimble. A string diagram calculus for predicate logic and C. S. Peirce’s system Beta. Preprint, 1998, revised 2000. <https://ncatlab.org/nlab/files/BradyTrimbleString.pdf>, 2000.
- [7] William Bricken. A deductive mathematics for efficient reasoning. Technical Report HITL-R-86-2, Human Interface Technology Laboratory, 1986.
- [8] William Bricken. What’s the difference? Contrasting boundary and Boolean algebras. Manuscript, October 2005. <http://iconicmath.com/mypdfs/bl-the-difference.080830.pdf>, 2005.
- [9] William Bricken. Boundary logic and alpha existential graphs. Manuscript dated 2006-01-23. <http://iconicmath.com/mypdfs/bl-aeg.060123.pdf>, 2006.
- [10] Pablo Donato. The flower calculus. In *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, volume 299 of *LIPIcs*, pages 5:1–5:24. Schloss Dagstuhl, 2024.
- [11] Nathan Haydon and Paweł Sobociński. Compositional diagrammatic first-order logic. In *Diagrammatic Representation and Inference (Diagrams 2020)*, volume 12169 of *LNCS*, pages 402–418. Springer, 2020.
- [12] Leon Henkin, J. Donald Monk, and Alfred Tarski. *Cylindric Algebras, Part II*, volume 115 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1985.
- [13] Edward V. Huntington. New sets of independent postulates for the algebra of logic, with special reference to Whitehead and Russell’s Principia mathematica. *Transactions of the American Mathematical Society*, 35(1):274–304, 1933.
- [14] Louis H. Kauffman. Robbins algebra. <http://homepages.math.uic.edu/~kauffman/Robbins.htm>.
- [15] Louis H. Kauffman and Arthur M. Collings. The BF calculus and the square root of negation. In *Laws of Form: A Fiftieth Anniversary*, Series on Knots and Everything, pages 253–283. World Scientific, 2023. Preprint arXiv:1905.12891.
- [16] Minghui Ma and Ahti-Veikko Pietarinen. Proof analysis of Peirce’s alpha system of graphs. *Studia Logica*, 105(3):625–647, 2017.
- [17] William McCune. Solution of the Robbins problem. *Journal of Automated Reasoning*, 19(3):263–276, 1997.
- [18] William McCune. Prover9 and Mace4, 2005–2010. Version 2009-11A. <https://www.cs.unm.edu/~mccune/prover9/>.
- [19] William McCune, Robert Veroff, Branden Fitelson, Kenneth Harris, Andrew Feist, and Larry Wos. Short single axioms for Boolean algebra. *Journal of Automated Reasoning*, 29(1):1–16, 2002.
- [20] Philip Meguire. Boundary algebra: A simpler approach to Boolean algebra and the sentential connectives. Working Papers in Economics 10/50, Department of Economics and Finance, University of Canterbury, 2010. <https://repec.canterbury.ac.nz/cbt/econwp/1050.pdf>.
- [21] J. Donald Monk. Nonfinitizability of classes of representable cylindric algebras. *Journal of Symbolic Logic*, 34(3):331–343, 1969.
- [22] J. Donald Monk. Leon Henkin and cylindric algebras. In *The Life and Work of Leon Henkin*, Studies in Universal Logic, pages 59–66. Birkhäuser, 2014. Author’s copy: <https://euclid.colorado.edu/~monkd/monk85.pdf>.
- [23] Naipmoro. lofmm: Laws of Form in Metamath. <https://naipmoro.github.io/lofmm/>.

- [24] Nicholas Smallbone, Moa Johansson, Koen Claessen, and Maximilian Algehed. Quick specifications for the busy programmer. *Journal of Functional Programming*, 27:e18, 2017.
- [25] G. Spencer-Brown. *Laws of Form*. Allen and Unwin, London, 1969.
- [26] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002. pp. 816–818.

INDEPENDENT RESEARCHER

Email address: `soobtoob@gmail.com`