Semantical Investigations on Non-classical Logics with Recovery Operators: Negation

David Fuenmayor[1,2][0000−0002−0042−4538]

[1] AI Systems Engineering, Otto-Friedrich-Universität Bamberg, Germany
`david.fuenmayor@uni-bamberg.de`
[2] Mathematics and Computer Science, Freie Universität Berlin, Germany

Abstract. We investigate mathematical structures that provide natural semantics for families of (quantified) non-classical logics featuring special unary connectives, known as *recovery operators*, that allow us to 'recover' the properties of classical logic in a controlled manner. These structures are known as *topological Boolean algebras*, which are Boolean algebras extended with additional operations subject to specific conditions of a topological nature. In this study we focus on the paradigmatic case of negation. We demonstrate how these algebras are well-suited to provide a semantics for some families of paraconsistent *Logics of Formal Inconsistency* and paracomplete *Logics of Formal Undeterminedness*. These logics feature recovery operators used to earmark propositions that behave 'classically' when interacting with non-classical negations. Unlike traditional semantical investigations, which are carried out in natural language (extended with mathematical shorthand), our formal meta-language is a system of higher-order logic (HOL) for which automated reasoning tools exist. In our approach, topological Boolean algebras are encoded as algebras of sets via their Stone-type representation. We use our higher-order meta-logic to define and interrelate several transformations on unary set operations, which naturally give rise to a *topological cube of opposition*. Additionally, our approach enables a uniform characterization of propositional, first-order, and higher-order quantification, including restrictions to constant and varying domains. With this work, we aim to make a case for the utilization of automated theorem proving technology for conducting computer-supported research in non-classical logics. All the results presented in this paper have been formally verified, and in many cases obtained, using the Isabelle/HOL proof assistant.

Keywords: Higher-order Logic · Logics of Formal Inconsistency and Undeterminedness · Non-classical Negation · Quantifiers · Recovery Operators · Shallow Semantical Embeddings

1   Introduction

From the perspective of computer science and artificial intelligence (e.g., knowledge representation) it is natural to seek 'non-classical' logical formalisms that accommodate reasoning under real-world conditions, such as the presence of partial or contradictory information. For instance, by adopting an epistemic approach towards paraconsistency and paracompleteness in logic (e.g., [19]), we can view these as coping mechanisms for the inevitable occurrence of 'epistemically suboptimal' situations, where cognizers might encounter incomplete and/or contradictory evidence regarding the truth or falsity of propositions. This epistemic approach underscores the need to augment a logic with expressive means to qualify the (quality of) evidence for individual propositions, and to discern which among them can be designated 'safe' for classical reasoning. Several researchers suggestively refer to object-logical connectives fulfilling this role as *recovery operators* [17]. These are paradigmatically instantiated in the families of paraconsistent *Logics of Formal Inconsistency* (LFIs; introduced in [18], cf. also [15,17]) and paracomplete *Logics of Formal Undeterminedness* (LFUs; introduced in [40], cf. also [17]).

On a related note, spurred by ongoing efforts to implement symbolic capabilities in AI systems, particularly with respect to automated legal and normative reasoning (see e.g. [9,27] and references therein), a specific technique, known as *shallow semantical embedding* [6,10], has been developed with the intent of repurposing automated theorem proving technology for classical first- and higher-order logic to provide *practically effective*[3] automation for families of *quantified* non-classical logics. This technique introduces families of *object logics* by encoding their connectives as terms of a fully formal *meta-logic*, in which their truth conditions are stated and semantical constraints are axiomatized. This approach aligns closely with the original idea as proposed by Tarski [47], and which has influenced several generations of logicians, for example, by motivating the development of model theory. In contrast to traditional model-theoretical investigations that use natural language (extended with mathematical shorthand) as a meta-language, the meta-logic used in shallow semantical embeddings is a system of higher-order logic (HOL), extending Church's simple theory of types [20,11], which lies at the heart of many modern automated theorem provers and interactive proof assistant systems [8]. This approach has been utilized for interactive (meta-)theoretical investigations in (and about) quantified modal and non-classical logics using mathematical proof assistants, in particular Isabelle/HOL [43]. Moreover, this research has been applied to the logico-pluralistic, computer-supported formalization and analysis of philosophical arguments and normative theories (cf. [27,9,6] and references therein).

Building upon prior research on shallow semantical embeddings, we present in this work a class of algebras aimed at providing natural semantics for families of (quanti-

---

[3] Effective from a pragmatic, best-effort AI engineering perspective, meaning performing adequately in the challenging (predominantly *undecidable*) tasks encountered in contemporary AI. While theoretical computability and complexity issues can significantly influence the design of certain safety-critical systems (e.g., in cybersecurity), they cannot guarantee (nor deter) practical success of automated reasoning tools. A good example of this is the impressive performance of SAT-solvers (and more recently SMT-solvers) in formal verification, where they routinely solve complex problems (e.g., involving hundreds of variables) otherwise deemed 'intractable' by (worst-case) theoretical analysis.

fied) non-classical logics featuring recovery operators. These structures are referred to as *topological Boolean algebras* (TBAs).[4] TBAs are Boolean algebras extended with additional operations that satisfy axiomatic conditions of a topological nature. In this study we focus on the unary case and refer to these additional unary operations as *operators*. Traditional examples of operators in logic thus include (unary) modalities and negations. Furthermore, we concentrate on the paradigmatic case of negation (and corresponding recovery operators). We show how TBAs are aptly suited to provide semantics for certain families of (paraconsistent) LFIs and (paracomplete) LFUs.

We adopt a natural approach towards the shallow semantical embedding of TBAs, in which they become encoded as algebras of sets (via their Stone-type representation). We utilize our higher-order meta-logic to define and interrelate certain special transformations on operators, such that they naturally give rise to a commutative diagram in the form of a *topological cube of opposition*. We also discuss how our approach enables a uniform characterization of propositional, first-order, and higher-order quantification, including restrictions to constant and varying domains.

With this work we also want to make a case for the utilization of formal meta-languages and automated reasoners for doing computer-supported research in non-classical logics. In fact, all presented results have been formally verified, and in many cases obtained, using the Isabelle/HOL proof assistant [43]. Readers interested in examining the Isabelle/HOL sources corresponding to the results discussed in this paper are encouraged to visit our GitHub repository[5] where they can access the most recent version of our ongoing formalization work. Many of the results discussed here were previously presented at the 3rd International Conference on Non-Classical Modal and Predicate Logics (NCMPL 2021).

In the following section §2 we have made an effort to present some relevant background material in an uniform manner. In particular, much of the material on TBAs has been gathered from various sources scattered throughout the literature.[6] This material will help to establish the required conceptual framework for the discussion in §3, where we use the shallow semantical embedding approach to develop a theory of unary operations (operators) on top of Boolean algebras encoded as algebras of sets. We use this theory to provide uniform semantics for families of *quantified* LFIs and LFUs via TBAs. We conclude the paper in §4.

---

[4] We were inspired to choose this term by Rasiowa & Sikorski's monograph *The Mathematics of Metamathematics* [44]. McKinsey & Tarski's *The Algebra of Topology* [41] can be considered the seminal work investigating this kind of structures, where they are termed *closure algebras*.

[5] Visit `https://github.com/davfuenmayor/topological-semantics`. This repository hosts the development version of the corresponding *Archive of Formal Proofs* [26] entry.

[6] The early work on the axiomatic foundations of topology by Kuratowski [38,39] and Zarycki [48,49,50] deserves special mention here. We find it rather unfortunate, especially for logicians, that this way of introducing topology has not received the recognition it deserves. The interested reader can now find English translations of Zarycki's works which have been recently uploaded on the web (thanks to Mark Bowron). See `https://www.researchgate.net/scientific-contributions/Miron-Zarycki-2016157096`.

2   Conceptual Preliminaries

2.1   Paraconsistent and Paracomplete Logics

Logics featuring non-classical negation-like operators are not only of philosophical interest but also have important applications in computer science and artificial intelligence. In particular, they aid in the development of robust knowledge representation and reasoning techniques that can operate under conditions of incomplete and contradictory information.

A logic is termed *paraconsistent* when it features a negation operator ($\neg$) for which the principle of *ex contradictione (sequitur) quodlibet* (ECQ), also colloquially referred to as 'explosion', does not hold: not everything follows from a contradiction $A \land \neg A$. Dually, paracomplete logics feature a negation for which the principle of *tertium non datur* (TND), aka. 'excluded middle', is not valid: formulas of the form $A \lor \neg A$ are not tautologies. Intuitionistic logic, for instance, is a well-known paracomplete logic (out of infinitely many) and dual-intuitionistic logic is paraconsistent. Broadly speaking, paraconsistent logics can be said to 'tolerate contradictions', while paracomplete logics do not 'require exhaustiveness' of available information or evidence.[7]

It is known that any unary operation that validates both ECQ and TND (wrt. the Boolean $\land$ & $\lor$ connectives) collapses into classical negation. Thus, whenever we require a 'weaker' negation (upon a Boolean base), which invalidates some logical property of its classical counterpart, we are compelled to relinquish either TND or ECQ (or sometimes both). Consequently, our logic will inevitably become either paracomplete or paraconsistent (or sometimes both).[8]

Fig. 1 shows a suggestive pictorial representation of the well-known behavior of the classical negation, where the notation $|A|$ represents the semantical denotation of some sentence A as its so-called 'truth-set', i.e., the set of situations (aka. 'worlds') in which A happens to be true. As illustrated, classical logic makes the assumption that we always have complete and non-contradictory information or evidence as to whether A or its negation $\neg A$ is the case. It is instructive to compare Fig. 1 with the two (mutually dual) situations depicted in Fig. 2 for non-classical negations. As illustrated, employing paraconsistent negations we can represent *truth-gluts*, i.e., situations in which, for some A, both it and its negation $\neg A$ are the case (e.g., when we have contradictory information about A). Analogously, employing paracomplete negations we can represent *truth-gaps*, i.e., situations in which neither A nor its negation $\neg A$ is the case (e.g., when we have only partial information as to whether A holds).

---

[7] Note that we assume an epistemic interpretation of paraconsistency and paracompleteness (as in e.g. [19]). This is justified by our ongoing efforts in applying non-classical logics in areas of AI like knowledge representation and reasoning [27,9].

[8] Logics that are both paraconsistent and paracomplete are referred to as *paradefinite* (or *paranormal*) in the literature. We direct the reader to [15] for a comprehensive discussion on the theory and applications of paraconsistent and other non-classical logics.
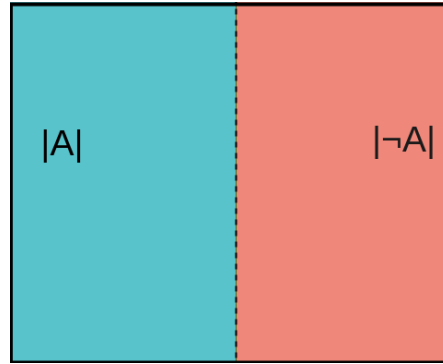
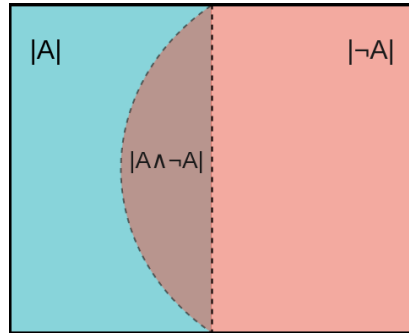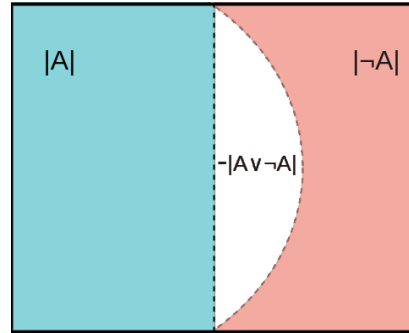Fig. 1: Suggestive pictorial representation of classical negation.

(a) Paraconsistency gives rise to *truth-gluts*

(b) Paracompleteness gives rise to *truth-gaps*

Fig. 2: Suggestive pictorial representation of non-classical negations.

Thus, in our approach, paraconsistency and paracompleteness can be seen as dual notions.[9] From a logical point of view:

$$(\text{paraconsistency}) \quad A \wedge \neg A \nvdash \bot$$
$$\top \nvdash A \vee \neg A \quad (\text{paracompleteness}).$$

As previously mentioned, given Boolean conjunction and disjunction, it becomes necessary to relinquish either TND or ECQ (or both) for any property of negation to become 'negotiable'. There are many compelling reasons to feature a 'weak' negation in a logical system, and these have been thoroughly discussed in the literature on non-classical logics (see e.g. [15] and the references therein). We have listed several properties of negation in Table 1 that have been the subject of our investigations (see [26]).

description	schema
'weak' TND	$\neg B \vdash A \vee \neg A$
'weak' ECQ	$A \wedge \neg A \vdash \neg B$
law of non-contradiction	$\vdash \neg(A \wedge \neg A)$
de Morgan 1	$\neg(A \vee B) \dashv\vdash \neg A \wedge \neg B$
de Morgan 2	$\neg(A \wedge B) \dashv\vdash \neg A \vee \neg B$
double negation	$A \dashv\vdash \neg\neg A$
'weak' double negation	$\vdash A \iff \vdash \neg\neg A$
contraposition 1	$A \to B \dashv\vdash \neg B \to \neg A$
contraposition 2	$A \to \neg B \dashv\vdash B \to \neg A$
'weak' contraposition 1	$A \vdash B \iff \neg B \vdash \neg A$
'weak' contraposition 2	$A \vdash \neg B \iff B \vdash \neg A$
disjunctive syllogism	$A \vee B \dashv\vdash \neg A \to B$

Table 1: Some 'negotiable' properties of negation. Note that each direction ($\vdash$ resp. $\dashv$ or $\implies$ resp. $\impliedby$) is a different property that should be considered separately.

As mentioned previously, one of our aims has been to facilitate automated reasoning with *non-classical* logical formalisms. The task of designing a custom logical system for a particular modeling problem (e.g., in AI knowledge representation) requires exploring a large space of logical inferences for different combinations of properties of the connectives in a target logic. Even for a single unary negation(-like) operation (cf. Table 1) the space to be explored is too vast for a systematic and *reliable* exploration using traditional pen-and-paper proof methods.[10] In our approach, state-of-the-art automated theorem provers and (counter)model generators can be harnessed to handle most of the

---

[9] Such a duality naturally follows from our consideration of lattice-based semantics for families of paraconsistent and paracomplete logics. As we will explore later, this frequently results in 'dual pairs' of characterizations and other meta-theoretical results.

[10] Some readers, being skilled logicians themselves, might feel capable of reproducing or verifying by hand all the results presented in our formalization work (see [26]) for each combination of assumptions. The questions are rather: do they truly have the time? how reliable are those proofs and who will check them? The reader might possess dependable intuitions concerning

proof and refutation heavy lifting. This way, we can systematically explore the *minimal* sets of semantic conditions that are necessary for interesting (meta)theorems to hold.[11]

2.2   Recovery Operators

Surprisingly (or not) paraconsistent and paracomplete logics are not necessarily 'weaker'[12] than classical logic. In fact, some paraconsistent systems in the literature are strictly more expressive than classical logic: they embed the latter as a subsystem. Among such systems we have the families of paraconsistent Logics of Formal Inconsistency (LFIs) [18,15,17] and paracomplete Logics of Formal Undeterminedness (LFUs) [40,17]. LFIs and LFUs feature additional unary connectives ($\circ$ and $\star$, respectively) allowing us to recover classical properties in a 'sentence-wise' fashion. Following [17] we refer to them as *recovery operators*. Their modus operandi is described below.[13]

LFIs, being paraconsistent, do not validate ECQ (aka. 'principle of explosion'), i.e.

$$A, \neg A \nvdash \Delta \,.$$

Instead, LFIs validate the so-called 'principle of gentle explosion', namely[14]

$$\circ A, A, \neg A \vdash \Delta \,.$$

The property above suggests reading the LFIs' operator $\circ$ as a kind of *consistency operator*, so that the formula $\circ A$ reads as "A is consistent" [18]. Thus, under the assumption that (the evidence for) A is consistent, i.e. $\circ A$, we have that from $A \wedge \neg A$ anything does follow.

LFUs, being paracomplete, do not validate the law of excluded middle (TND), i.e.

$$\Gamma \nvdash A, \neg A \,.$$

Dually to LFIs, LFUs validate instead

$$\Gamma \vdash \star A, A, \neg A \quad \text{i.e.} \quad \Gamma, \star A \vdash A, \neg A$$

where $\star A = -\star A$ for classical negation '$-$' (which is in fact definable inside the LFIs/LFUs considered here, as discussed below). Analogously, Marcos [40] proposes to read $\star A$ as "A is determined". Thus, assuming that (the evidence for) A is determined (i.e. complete or exhaustive), we will always have that $A \vee \neg A$ holds.

---

classical, intuitionistic, or even some well-known systems of paraconsistent logic, but these intuitions won't help (in fact they might hinder) in deriving or verifying proofs in other, more exotic systems.

[11] See [29] for an exemplary application of this methodology in topology.

[12] Weak(er) in the sense of validating less theorems, licensing less inferences, and so on.

[13] We assume a fully *structural* setting (in Gentzen's sense), and thus the comma-separator ',' in a sequent behaves as a conjunction (resp. disjunction) on the left- (resp. right-) hand-side.

[14] In the literature on LFIs [18,15] an additional restriction is considered, in order for this principle to hold in a non-trivial way: there exist formulas α, β such that $\circ\alpha, \alpha \nvdash \beta$ and $\circ\alpha, \neg\alpha \nvdash \beta$.

Drawing upon these intuitive readings we can think of the operators $\circ$ and $\star$ as sort of 'quality seals' for propositions. For example, they can offer insights into the 'trust-worthiness' level of evidential support for propositions: $\circ A$ indicates that the evidence for A is non-contradictory, while $\star A$ suggests that the evidence for A is non-partial. We refer the reader to [19] for a discussion of the epistemic interpretation of LFIs and LFUs.

Enhancing a logic with recovery operators such as $\circ$ and $\star$ not only serves to recover the properties of ECQ and TND as shown above. In fact, other properties of negation can be also be recovered this way. For example, in the LFI system **mbC** [15] contraposition (among others) is generally not valid, e.g.

$$A \rightarrow B \nvdash_{\mathbf{mbC}} \neg B \rightarrow \neg A\,.$$

From this we can correctly infer that the **mbC** negation ($\neg$) is indeed very 'weak'. However, we shall not extend this judgment to the whole logical system. On the one hand, we can always recover classical properties in a sentence-wise fashion by employing the consistency operator $\circ$, thus for the case above we have

$$\circ B, A \rightarrow B \vdash_{\mathbf{mbC}} \neg B \rightarrow \neg A\,.$$

On the other hand, the additional expressivity obtained via the recovery operator $\circ$ allows us to define, in systems extending **mbC**, a fully classical negation inside the logic as $-A = A \rightarrow \bot$, where $\bot$ is any formula of the form $B \wedge \neg B \wedge \circ B$. In this sense, we can say that, in fact, those LFIs 'extend' or 'embed' classical logic. Of course, an analogous (dual) argument can be made for LFUs too.

We finish this section by noting that recent work [16] has provided an algebraic semantics for extensions of the LFI system **mbC** that satisfy the property of *replacement* (of provable equivalents). The minimal system in this new family has been suggestively called **RmbC**. The semantics provided in that work for **RmbC** and its extensions are based upon so-called *Boolean algebras with LFI operators* (BALFIs). The algebraic semantics introduced in the present paper (based on topological Boolean algebras, cf. §3) has, in fact, grown out of an effort to generalize the neighborhood structures introduced as representations of BALFIs in [16, §5].

In the sequel we will always assume that our non-classical logical systems are *self-extensional*, i.e., they satisfy the replacement property.

2.3   Topological Boolean Algebras

We introduce in §3 a family of mathematical structures that, we argue, provide a natural semantics for some families of LFIs and LFUs, and for which we employ the umbrella term: *topological Boolean algebras* (TBAs).[15] We characterize TBAs rather informally

---

[15] We have chosen this name inspired by Rasiowa & Sikorski [44] who exemplarily employ structures of this kind to provide semantics for non-classical logics. However, as will be seen, we generalize its meaning to cover not only ('weakenings' of) closure and interior algebras, but also Boolean algebras featuring other operations of a somewhat topological character. The term TBAs, in our sense, is left deliberately vague. We make no attempt to define it.

as Boolean algebras extended with one or more operations satisfying some (but not necessarily all) of the properties traditionally associated with well-known topological operations on sets. As for the unary case, paradigmatic cases of such operations (operators) are the topological *closure* and its dual the *interior* [38]. Moreover, other set-operations can also be interpreted as topological, in particular the *border*, *frontier* and *exterior* [48], as well as the *derived-set* (aka. *derivative*) [38,50,41], Hausdorff's *residue* [34] and Cantor's *coherence* [49]. They all count as operators in our sense of the word.

The first and most paradigmatic examples of TBAs in the literature are *closure algebras*, i.e., Boolean algebras extended with a (topological) closure operator. They were in fact the structures studied in the seminal work by McKinsey & Tarski [41]. The corresponding closure operator and its dual (interior) were famously axiomatized by Kuratowski [38].

Readers familiarized with basic topology will recall the common definition of topological spaces as algebras of open (resp. closed) sets, for which an interior (resp. closure) operation can be defined in the usual way. However, only a few might be familiarized with the definition of topological spaces as closure algebras, as presented, e.g., by McKinsey & Tarski [41]. Intuitively, the equivalence of both characterizations should not come as a surprise, since, together with the simple 2-element lattice $\{\top, \bot\}$, algebras of sets (aka. *fields of sets*) provide the paradigmatic examples of Boolean algebras.[16]

We recall below some characterizations of topological operators that will be relevant in our discussion in §3. They are intended as additional unary operations on top of a Boolean algebra with carrier $\mathfrak{A}$, signature $\langle \wedge, \vee, -, \top, \bot \rangle$, and ordered by $\leq$. We provide some names and mnemonics for their axiomatic conditions (ADDItivity, IDEMpotence, etc.) to facilitate references.

Definition 1 (Closure Operator). A unary operation C is called a (Kuratowski) *closure operator* when, for arbitrary $A, B \in \mathfrak{A}$, it is

(C1)	additive (ADDI) :	$C(A \vee B) = C(A) \vee C(B)$
(C2)	expansive (EXPN) :	$A \leq C(A)$
(C3)	normal (NORM) :	$C(\bot) = \bot$
(C4)	idempotent (IDEM) :	$C(C(A)) = C(A)\,.$

A Boolean algebra extended with a closure operator is called a *closure algebra*.

Definition 2 (Interior Operator). A unary operation $\mathcal{I}$ is called an *interior operator* when, for arbitrary $A, B \in \mathfrak{A}$, it is

(I1)	multiplicative (MULT) :	$\mathcal{I}(A \wedge B) = \mathcal{I}(A) \wedge \mathcal{I}(B)$
(I2)	contractive (CNTR) :	$\mathcal{I}(A) \leq A$
(I3)	dual-normal (DNRM) :	$\mathcal{I}(\top) = \top$
(I4)	idempotent (IDEM) :	$\mathcal{I}(A) = \mathcal{I}(\mathcal{I}(A))\,.$

A Boolean algebra extended with an interior operator is called an *interior algebra*.

---

[16] This has of course its roots in the celebrated representation theorems by Birkhoff and Stone.

Closure and interior operators are said to be *dual*, since given one of them, say C, we can define the other, say $\mathcal{I}(A) \overset{\text{def}}{=} -C(-A)$ for any A. Let us make this relationship official by defining the dual $(\cdot)^{\text{d}}$ of an operator φ as $\varphi^{\text{d}}(A) \overset{\text{def}}{=} -\varphi(-A)$.

It is well known that closure/interior algebras and topological spaces are two sides of the same coin. This insight has been leveraged, at least since the seminal work by McKinsey & Tarski in the 1940's [41], to provide topological semantics for intuitionistic and modal logics (see [23] for a brief survey). In fact, by employing different subsets of the closure conditions listed above (C1-4) we can also characterize some interesting operators, to name just a few:

- The conditions C1 and C3 (dually: I1 and I3) suffice to characterize a modal *possibility* (dually: *necessity*) operator.[17]
- The conditions C1, C2 and C3 (leaving out idempotence: C4) characterize a preclosure (aka. Čech closure) operator.
- The ubiquitous Moore (aka. hull) closure is characterized by the right-to-left direction ($\geq$) in C1 (equivalent to monotonicity), together with C2 and C4.
- Another topological operator, the derived-set (aka. derivative) operation [38,50,41] can be characterized by C1, C3, and the left-to-right direction ($\leq$) in C4.

In instances like those above where we utilize only a subset of the Kuratowski closure conditions (C1-4) to constrain a given unary operation, say C, we might rigorously refer to it, for example, as a 'generalized closure(-like) operator'. Analogous designations could be devised for unary operations satisfying only a subset of the interior conditions (I1-4) or the other sets of conditions to be introduced later. To circumvent such contrived terminologies, we will adopt the terms 'closure', 'interior', etc., more loosely, so they also apply to 'weaker' operators. We will let the context determine how operators are referred to, as long as it does not lead to confusion.

Bridging between the notions of closure and interior we find the notion of *exterior* (of a set), which has been axiomatized by Zarycki [48] as follows.

Definition 3 (Exterior Operator). A unary operation $\mathcal{E}$ is called an *exterior operator* when, for arbitrary $A, B \in \mathfrak{A}$, it satisfies

$$(E1) \qquad \mathcal{E}(A \vee B) = \mathcal{E}(A) \wedge \mathcal{E}(B)$$

$$(E2) \qquad \mathcal{E}(A) \leq -A$$

$$(E3) \qquad \mathcal{E}(\bot) = \top$$

$$(E4) \qquad \mathcal{E}(-\mathcal{E}(A)) = \mathcal{E}(A).$$

We may call a Boolean algebra extended with an exterior operator an *exterior algebra*.

Note that the axiomatic conditions for the exterior operator display a certain antagonism towards those for interior and closure (e.g., E1 will be referred to as 'anti-additivity'

---

[17] Cf. Jónsson & Tarski's *Boolean Algebras with Operators* [37]. This seminal work provided, in today's terminology, an algebraic semantics for (extensions of) normal modal logic **K**. Observe that condition I2 (C2) and the left-to-right direction ($\leq$) of I4 (C4) correspond to the (duals of) modal axioms T and 4 respectively. Hence taking all Kuratowski conditions together corresponds to axiomatizing modal logic **S4**.

later in §3). From another perspective, these concepts are interrelated since, given a closure C (resp. an interior I), an exterior operator $\mathcal{E}$ can be defined as $\mathcal{E}(A) \overset{\text{def}}{=} -C(A)$ (resp. $\mathcal{E}(A) \overset{\text{def}}{=} I(-A)$). Therefore, as with interior operators, exterior operators can be readily obtained by composing closure operators with the complement. In fact, Kuratowski [38] famously demonstrated that the number of operators achievable in this manner is limited to 14.[18]

Other topological operators axiomatized by Zarycki in [48] were the *border* and the *frontier* (aka. *boundary*).[19] The former, in particular, will play an important role in our investigations on recovery operators in §3.7.

Definition 4 (Border Operator). A unary operation $\mathcal{B}$ is called a *border operator* when, for arbitrary $A, B \in \mathfrak{A}$, it satisfies[20]

$$(\text{B1}) \qquad A \wedge B \wedge \mathcal{B}(A \wedge B) = A \wedge B \wedge (\mathcal{B}(A) \vee \mathcal{B}(B))$$

$$(\text{B2}) \qquad \mathcal{B}(A) \leq A$$

$$(\text{B3}) \qquad \mathcal{B}(\top) = \bot$$

$$(\text{B4}) \qquad \mathcal{B}(-\mathcal{B}(-A)) \leq A\,.$$

We may call a Boolean algebra extended with a border operator a *border algebra*.

Definition 5 (Frontier Operator). A unary operation $\mathcal{F}$ is called a *frontier operator* when, for arbitrary $A, B \in \mathfrak{A}$, it satisfies

$$(\text{F1}) \qquad A \wedge B \wedge \mathcal{F}(A \wedge B) = A \wedge B \wedge (\mathcal{F}(A) \vee \mathcal{F}(B))$$

$$(\text{F2}) \qquad \mathcal{F}(-A) = \mathcal{F}(A)$$

$$(\text{F3}) \qquad \mathcal{F}(\bot) = \bot$$

$$(\text{F4}) \qquad \mathcal{F}(\mathcal{F}(A)) \leq \mathcal{F}(A)\,.$$

We may call a Boolean algebra extended with a frontier operator a *frontier algebra*.

From the conditions above we can discern that the border ($\mathcal{B}$) and frontier ($\mathcal{F}$) operators are related in some sense, albeit they are still quite distinct. For instance, conditions B1 and F1 are identical. This condition has been dubbed in §3.4 (somewhat awkwardly) 'anti-multiplicativity relative to $(A \wedge B)$'. However, condition B2, which is identical to I2 (contractiveness), differs from condition F2 (symmetry wrt. complement). Condition F3 is identical to C3 (normality), while B3 has a reminiscent form (it has been dubbed,

---

[18] Of course, this result hinges on assuming (all?) Kuratowski closure conditions.

[19] They have been called 'bord' and 'frontière', respectively, in the early writings (in French) of Zarycki, Kuratowski, and other mathematicians of the era. The term 'boundary' is somewhat more modern and widespread, but we abstain from using it here, as it is commonly employed for another related concept.

[20] We note that the axiomatization originally introduced by Zarycki [48] features only three conditions, since it collapses our B1 and B2 into one equivalent formula: $\mathcal{B}(A \wedge B) = (A \wedge \mathcal{B}(B)) \vee (B \wedge \mathcal{B}(A))$. Our formulation here is not only more explicit but also makes it easier to interrelate conditions for different operators with a finer granularity.

awkwardly again, 'anti-dual-normality' in §3.4). Finally, F4 corresponds to the $\leq$ direction of C4 (idempotence), while B4 has a quite different form.[21] These sets of axiomatic conditions and their interrelationships will be systematically explored in §3.4.

We have quoted above the famous result by Kuratowski [38], which states that a maximum of 14 different operators can be defined by composing the complement ($-$) with a (fully axiomatized) closure operator C (and, by extension, with I and $\mathcal{E}$). In the same spirit, Zarycki [48] demonstrates that any operator obtained via composition of the complement with the frontier operator ($\mathcal{F}$) must coincide with one of the following six operations, which given A return respectively: $A, \mathcal{F}(A), \mathcal{F}(\mathcal{F}(A)), -A, -\mathcal{F}(A), -\mathcal{F}(\mathcal{F}(A))$. In contrast, Zarycki [48] also indicates that an infinite number of operators can be obtained in this manner when considering the border operator ($\mathcal{B}$).[22]

Zarycki [48] also shows that closure, interior, exterior, border and frontier operators are all inter-definable, i.e., having introduced one of them as primitive we can employ it to define the others as follows (see Table 2 and also Fig. 3 for a suggestive illustration):

define\using	C	I	$\mathcal{E}$	$\mathcal{B}$	$\mathcal{F}$
$C(A)$	$-$	$I^{\mathrm{d}}(A)$	$-\mathcal{E}(A)$	$A \vee \mathcal{B}(-A)$	$A \vee \mathcal{F}(A)$
$I(A)$	$C^{\mathrm{d}}(A)$	$-$	$\mathcal{E}(-A)$	$A \wedge -\mathcal{B}(A)$	$A \wedge -\mathcal{F}(A)$
$\mathcal{E}(A)$	$-C(A)$	$I(-A)$	$-$	$-A \wedge \mathcal{B}^{\mathrm{d}}(A)$	$-A \wedge -\mathcal{F}(A)$
$\mathcal{B}(A)$	$A \wedge C(-A)$	$A \wedge -I(A)$	$A \wedge \mathcal{E}^{\mathrm{d}}(A)$	$-$	$A \wedge \mathcal{F}(A)$
$\mathcal{F}(A)$	$C(A) \wedge C(-A)$	$-(I(A) \vee I(-A))$	$-(\mathcal{E}(A) \vee \mathcal{E}(-A))$	$\mathcal{B}(A) \vee \mathcal{B}(-A)$	$-$

Table 2: Inter-definitions between topological operators.

It is worth noting that inter-definitions involving *only* C, I and $\mathcal{E}$ obtain already without assuming any restriction. By contrast, inter-definitions involving any of $\mathcal{B}$ or $\mathcal{F}$ always assume the second Kuratowski condition (or its respective counterpart: C2, I2, E2, B2 or F2). For example, recalling Table 2, converting an operator, say $\mathcal{B}$, into an interior $I_{\mathcal{B}}$ (entry $\langle 4, 2 \rangle$) and then back to $\mathcal{B}$ (entry $\langle 2, 4 \rangle$) requires assuming B2.

As we have seen, after extending a Boolean algebra with any of the topological operators discussed above (C, I, $\mathcal{E}$, $\mathcal{B}$, $\mathcal{F}$), we obtain a TBA where all other operators are definable. Moreover, as will be shown in §3, several other operators can be defined by employing, among others, function composition, for which we employ the notation $(\varphi \circ \psi)(A) \stackrel{\text{def}}{=} \varphi(\psi(A))$ and write φ^{n+1} for φ composed with itself n-times.

Furthermore, Zarycki has also shown in [48] how to inter-derive all corresponding axiom systems for the above operators on top of algebras of sets. We can axiomatize, e.g., an operator $\mathcal{F}$ with the frontier conditions F1-4 (thus obtaining a frontier algebra) and derive, as theorems, the closure conditions C1-4 for a closure operator defined as

---

[21] Border operators are in fact idempotent, i.e., $\mathcal{B}(\mathcal{B}(A)) = \mathcal{B}(A)$; this follows already from B1 and B2. Furthermore, B4 is related to a variant of what we refer to as 'anti-idempotency' in §3.4.

[22] This is because there exist sets with a transfinite family of residues, where the residue of a set A corresponds to $\mathcal{B}(-\mathcal{B}(-A))$; cf. Hausdorff [34].
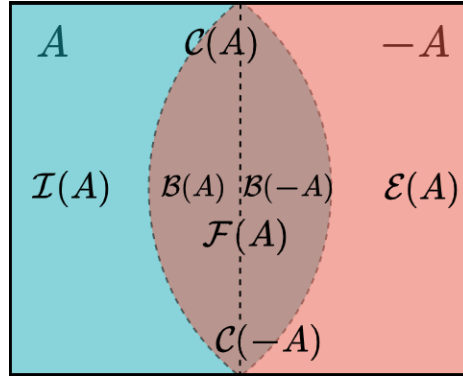
Fig. 3: Suggestive pictorial representation of topological operators and their interrelations. The blue, resp. red, areas correspond to $C(A)$, resp. $C(-A)$; they overlap in the brownish area corresponding to $\mathcal{F}(A)$ ($= \mathcal{F}(-A)$), itself partitioned into $\mathcal{B}(A)$ and $\mathcal{B}(-A)$ (corresponding in a sense to the *truth-gluts* and *truth-gaps* in Fig. 2). Observe also that $\mathcal{E}(A)$ corresponds to $\mathcal{I}(-A)$, and thus to $-C(A)$.

$C(A) \stackrel{\text{def}}{=} A \vee \mathcal{F}(A)$. We have employed in [26] automated tools to elicit *minimal* sets of conditions under which these correspondences hold. We will revisit this issue in §3.4.

Fixed points of topological operators are also interesting to consider. We employ below the notation $\mathtt{fp}(\varphi, A) \stackrel{\text{def}}{=} \varphi(A) = A$ (read: "A is a fixed point of φ"), recalling that fixed points of C, resp. $\mathcal{I}$, are called *closed*, resp. *open*, in the literature. In a similar vein, we can consider the fixed points of the operators discussed above and observe that they satisfy interesting topological properties (with operators inter-defined as in Table 2).

- A is *open* iff its border is the least element ('bottom'): $\mathtt{fp}(\mathcal{I}, A)$ iff $\mathcal{B}(A) = \bot$.
- A is *closed* iff the border of its complement is 'bottom': $\mathtt{fp}(C, A)$ iff $\mathcal{B}(-A) = \bot$.
- A is *clopen* (both closed and open) iff its frontier is 'bottom': $\mathtt{fp}(C, A)$ and $\mathtt{fp}(\mathcal{I}, A)$ iff $\mathcal{F}(A) = \bot$.
- $\mathcal{E}$ has no fixed points. However, fixed points of $\mathcal{E}^2$ correspond to *regular open* elements: $\mathtt{fp}(\mathcal{E}^2, A)$ iff $\mathtt{fp}(\mathcal{I} \circ C, A)$; dually, fixed points of $(\mathcal{E}^d)^2$ correspond to *regular closed* elements.
- Fixed points of $\mathcal{B}$ correspond to *boundary* elements (being analogous to the so-called 'boundary sets' in point-set topology): $\mathtt{fp}(\mathcal{B}, A)$ iff $\mathcal{I}(A) = \bot$.
- Fixed points of $\mathcal{B}^d$ correspond to *dense* elements: $\mathtt{fp}(\mathcal{B}^d, A)$ iff $C(A) = \top$.
- Fixed points of $\mathcal{F}$ correspond to those elements that are *closed* and *nowhere-dense*: $\mathtt{fp}(\mathcal{F}, A)$ iff $\mathtt{fp}(C, A)$ and $\mathcal{I}(C(A)) = \bot$. Conversely, *nowhere-dense* elements can be characterized as those whose closure is a fixed point of $\mathcal{F}$ or, alternatively, as the elements that are smaller than the frontier of their closure: $\mathcal{I}(C(A)) = \bot$ iff $\mathtt{fp}(\mathcal{F}, C(A))$ iff $A \leq \mathcal{F}(C(A))$.

Certainly, in the topological literature, it is always presupposed that operators satisfy all of the corresponding axiomatic (e.g., Kuratowski) conditions. A distinctive feature

of our approach is that these axiomatic conditions become 'negotiable',[23] allowing us to pose interesting 'reverse-mathematical' sort of questions involving the *minimal* set of conditions required, e.g., for the relationships listed above to hold. Some of these questions have been addressed in a topological context in previous work [29]. We argue that, in order to answer such questions *at scale*, we need to leverage computational tools like automated theorem provers and model generators. For instance, we employ the automated tools integrated into the Isabelle/HOL proof assistant. Automated theorem provers are utilized (via *Sledgehammer* [12]) to verify entailment and (counter)model generators (like *Nitpick* [13]) are employed to verify satisfiability and to find counter-examples to non-theorems (e.g., when some necessary assumption is lacking).

2.4   Degree-preserving Logical Consequence

Finally, having introduced our target classes of algebraic structures (TBAs), we discuss how they can serve to provide a semantics for families of non-classical logics such as LFIs and LFUs. We note that our target logics are assumed to be structural (in Gentzen's sense) and Tarskian. They also satisfy the property of replacement (of provable equivalents), i.e., they are self-extensional.

We begin with the classical connectives. As expected, conjunctions and disjunctions are interpreted by meets and joins ($\wedge$, $\vee$), respectively. We may also conveniently include 0-ary 'top' and 'bottom' connectives in our logic, interpreted by their algebraic counterparts ($\top$, $\bot$). Classical negation may or may not be part of our language; if it is, then it becomes interpreted by the complement operation ($-$). Implication is interpreted by the corresponding algebraic connective (or defined as shorthand, e.g., using classical negation and disjunction). We can optionally introduce 'modalities' in our language, which are to be interpreted by operators (e.g., $\Diamond$ interpreted as $\mathcal{C}$ and $\Box$ as $\mathcal{I}$).

A distinguishing feature of our approach is that non-classical negations (paraconsistent or paracomplete) and their corresponding recovery operators are interpreted by operators that are suitably interrelated. This will be discussed in §3.6-3.7.

Importantly, we interpret logical entailment using the algebra's ordering relation. Thus, e.g.,

$$A, B \vdash C \,(, D)$$

becomes interpreted as[24]

$$A \wedge B \leq C \,(\vee D)\,.$$

This approach towards characterizing logical consequence for (semi-)lattice-based algebraic semantics is well-known in the literature (e.g., in fuzzy logics), among others, under the name of *degree-preserving* consequence [14], and corresponds, from an algebraic point of view, to the standard notion of consequence employed in modal logic,

---

[23] As previously mentioned, this is even to the extent that qualifications like 'closure' or 'closed' might become challenging to justify. We stick to them, however.

[24] Recall that, since we assume a fully *structural* setting (in Gentzen's sense), the comma-separator ',' in a sequent behaves as a conjunction (disjunction) on the left- (right-) hand-side. Families of non-classical logics for which this assumption does not hold are called *substructural logics* in the literature. Their treatment lies beyond the scope of the present paper.

so-called *local* consequence. This can be contrasted with *truth-preserving* (aka. *global*) consequence, in which, e.g.,

$$A, B \vdash_g C \,(, D)$$

becomes interpreted as

$$A = \top \ \text{ and } \ B = \top \ \text{ implies } \ C = \top \ (\text{or } D = \top)\,.$$

Clearly, theorems coincide in both approaches, where

$$\vdash A \quad \text{iff} \quad A = \top \quad \text{iff} \quad \vdash_g A\,.$$

However, degree-preserving consequence is stronger than truth-preserving consequence. It is easy to see that $A \vdash B$ implies $A \vdash_g B$ but not vice versa.

A characteristic of the notion of degree-preserving consequence is that any logic featuring an intuitionistic (or classical) implication automatically satisfies the deduction theorem (wrt. that connective). Hence we have that

$$A \vdash B \quad \text{iff} \quad A \leq B \quad \text{iff} \quad A \to B = \top \quad \text{iff} \quad \vdash A \to B\,.$$

3   Semantical Investigations

The main results of our work are presented in this section, which is structured as follows: We begin in §3.1 by discussing the *shallow semantical embedding* approach, which we employ to develop a theory of unary operations (operators) on top of Boolean algebras, encoded in §3.2 as algebras of sets. In §3.3 we introduce some simple transformations between operators giving rise to an interesting commutative diagram in the form of an abstract 'cube of opposition'. We show in §3.4 how different sets of axiomatic conditions (à la Kuratowski) become 'translated' as we transform operators into each other (moving along the edges of the cube). In §3.5 we discuss some concrete operators, as featured in topological Boolean algebras (TBAs), and use them to instantiate a *topological cube of opposition*. We then show how non-classical (paraconsistent and paracomplete) negations (in §3.6) and recovery operators (in §3.7) can be defined in terms of topological operations and their transformations. Finally, we provide in §3.8 a uniform characterization of propositional, first-order, and higher-order quantification (restricted also to constant and varying domains) and discuss the Barcan formula (and its converse).

3.1   Higher-order Logic (HOL) as a Meta-Logic

We have mentioned previously that our target non-classical logics are to be introduced as a family of *object logics* by encoding logical connectives as terms in a *classical higher-order meta-logic*, in which their truth conditions are stated and semantic conditions axiomatized. This formal meta-logic is suggestively referred to as HOL in this article.[25]

---

[25] HOL is in fact an acronym for (classical) higher-order logic. It is traditionally employed to refer to a family of extensions of Church's simple type theory [11] commonly employed in interactive theorem proving, starting with the eponymous *HOL* proof assistant [31] and continuing with descendant systems like *HOL-Light* [33] and *Isabelle/HOL* [43].

HOL is a *logic of functions*. As a higher-order logic, HOL is a very expressive logical system that allows for quantification over predicate and function variables. In the context of automated reasoning, the term *higher-order logic* (in general) and HOL [32] (in particular) customarily refers to *classical* logical systems extending the simply typed λ-calculus, as originally introduced by Church [20], and often referred to as *simple type theory* [11].

To keep this article sufficiently self-contained we will succinctly introduce HOL's syntax and briefly motivate its semantics. We refer the interested reader to the literature on HOL semantics [35,3,7], proof systems and their automation [8].

As a typed logic, all terms of HOL get assigned a fixed and unique *type*, commonly written as a subscript (i.e., the term t_α has α as its type). The use of types in HOL aims at eliminating the paradoxes and inconsistencies commonly found in untyped systems. The set $\mathcal{T}$ of HOL's types is inductively generated from a set of base types $\mathcal{BT}$ and the function type constructor $\Rightarrow$ (written as a right-associative infix operator[26], see Fig. 4). Traditionally, the generating set $\mathcal{BT}$ is taken to include at least two base types, $\mathcal{BT} \supseteq \{\iota, o\}$, where ι is intuitively interpreted as the type of *individuals* and o as the type of Boolean *truth-values*. For instance, o, $o \Rightarrow o$, $\iota \Rightarrow \iota$ and $\iota \Rightarrow \iota \Rightarrow o$ are types. Further base types may be added to $\mathcal{BT}$ as needed.

$$\alpha, \beta \ ::= \ \tau \in \mathcal{BT} \ \mid \ \alpha \Rightarrow \beta$$
$$s, t \ ::= \ c_\alpha \in Const \ \mid \ x_\alpha \in \mathcal{V}ar \ \mid \ \left(\lambda x_\alpha. s_\beta\right)_{\alpha \Rightarrow \beta} \ \mid \ \left(s_{\alpha \Rightarrow \beta}\, t_\alpha\right)_\beta$$

Fig. 4: Formal grammar for HOL types ($\alpha, \beta \in \mathcal{T}$) and terms ($s, t \in \mathcal{L}$).

In contrast to traditional first-order logic, HOL is a logic of terms only (terms of type o are customarily referred to as 'formulas'). The set $\mathcal{L}$ of HOL's *terms* is inductively defined starting from a collection of typed constant symbols (*Const*) and typed variable symbols (*$\mathcal{V}ar$*) using the constructors *function abstraction* and *function application*, thereby obeying type constraints as indicated (see Fig. 4). Type subscripts and parentheses are usually omitted to improve readability, if obvious from the context or irrelevant. Moreover, we shall assume that *Const* contains the symbols $Q^\alpha_{\alpha \Rightarrow \alpha \Rightarrow o}$ (infix notation: $=^\alpha$) for each $\alpha \in \mathcal{T}$, which are interpreted as (primitive) *equality*. Logical symbols, including conjunction ($\wedge_{o \Rightarrow o \Rightarrow o}$), disjunction ($\vee_{o \Rightarrow o \Rightarrow o}$), material implication ($\longrightarrow_{o \Rightarrow o \Rightarrow o}$), negation ($\neg_{o \Rightarrow o}$), Boolean constants ($\mathtt{true}_o$, $\mathtt{false}_o$) and universal quantification for predicates over type α ($\Pi^\alpha_{(\alpha \Rightarrow o) \Rightarrow o}$) can be introduced as primitive constant symbols in *Const* (and thus given fixed intended interpretations), or, alternatively, they are often defined as abbreviations in terms of equality Q^α [11,3]. In any case, we refer to them as HOL's *logical connectives*. For convenience, *binder notation* is introduced for quantifiers, such that $\forall^\alpha x_\alpha. s_o$ can be used as an abbreviation for $\Pi^\alpha_{(\alpha \Rightarrow o) \Rightarrow o} \lambda x_\alpha. s_o$ for each $\alpha \in \mathcal{T}$. Additionally, *description* and *choice* operators may be added to the language [11,3].

Thus, HOL provides λ-notation as an expressive variable-binding mechanism to represent unnamed functions (*function abstraction*), which can also be employed to encode

---

[26] We can e.g. omit parentheses in the type $\alpha \Rightarrow (\beta \Rightarrow \gamma)$ but not in the type $(\alpha \Rightarrow \beta) \Rightarrow \gamma$.

predicates and sets via their *characteristic functions*. Note that, by construction, HOL syntax only admits functions that take one parameter; e.g., the function that adds 2 to a given number can be represented as $(\lambda y. 2 + y)$. Functions having two or more arguments must be encoded in HOL in terms of one-argument functions. In this case the values returned when applying these functions are themselves functions, which are subsequently applied to the next argument; e.g., the function that adds two numbers can be encoded as $(\lambda x. (\lambda y. x + y))$. This technique, introduced in [46], is commonly called *currying*. Hence, HOL can represent *partial application* for n-ary functions; e.g., the term $(\lambda y. 2 + y)$ can be used to represent the partial application of the binary function '+' to the argument '2'.

For our current purposes, we shall omit the detailed presentation of HOL semantics and of its numerous proof systems, and instead refer the interested reader to the literature (e.g., [7,8,11,3]). We restrict ourselves to mentioning that in set-based approaches to HOL's semantics each model is based upon a *frame*: a collection $\{\mathcal{D}_\alpha\}_{\alpha\in\mathcal{T}}$ of non-empty sets, called *domains* (for each type α). Since HOL is classical, $\mathcal{D}_o$ is restricted to a two-element set, say $\{T, F\}$, whereas $\mathcal{D}_\iota$ may have arbitrarily many elements (aka. individuals). Expectedly, the set $\mathcal{D}_{\alpha\Rightarrow\beta}$ consists of functions with domain $\mathcal{D}_\alpha$ and codomain $\mathcal{D}_\beta$. In so-called *standard models* [11], these domain sets are assumed to be *full* (i.e. they contain *all* functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$). Broadly speaking, the main idea is to define a so-called *denotation function* $(|\cdot|)$ that interprets each term s_α as an element $|s_\alpha|$ of $\mathcal{D}_\alpha$ (its *denotation*). As expected, a *denotation function* worthy of its name must respect the intended semantics of HOL as a *logic of functions*. Thus, $|s_{\alpha\Rightarrow\beta}\ t_\alpha| \in \mathcal{D}_\beta$ denotes the value of the function $|s_{\alpha\Rightarrow\beta}| \in \mathcal{D}_{\alpha\Rightarrow\beta}$ when applied to $|t_\alpha| \in \mathcal{D}_\alpha$, and $|\lambda x_\alpha. s_\beta| \in \mathcal{D}_{\alpha\Rightarrow\beta}$ denotes an (unique) 'appropriate' function from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$ (see e.g. [11, §2] for details on how to define this). It shall be noted that the (primitive) equality constants $\mathcal{Q}^\alpha_{\alpha\Rightarrow\alpha\Rightarrow o}$ (infix: $=^\alpha$) must be given a fixed intended interpretation in such a way that $|A_\alpha =^\alpha B_\alpha|$ denotes $T \in \mathcal{D}_o$ iff $|A_\alpha|$ is identical to $|B_\alpha| \in \mathcal{D}_\alpha$. The constants $\texttt{true}_o$, $\texttt{false}_o$ are interpreted by $T, F \in \mathcal{D}_o$ respectively. The quantifier constants Π^α are interpreted as 2nd-order predicates, i.e. $|\Pi^\alpha| \in \mathcal{D}_{(\alpha\Rightarrow o)\Rightarrow o}$, that take a predicate as an argument and return $T \in \mathcal{D}_o$ iff the predicate holds for *all* the objects in the domain $\mathcal{D}_\alpha$. It is worth noting that we assume *functional extensionality* in HOL [11,7]. Therefore, quantifiers can be defined as abbreviations using equality: $\Pi^\alpha \overset{\text{def}}{=} \lambda P_{\alpha\Rightarrow o}. (P =^{\alpha\Rightarrow o} \lambda x_\alpha. \texttt{true}_o)$.

Logic folklore has it that, as a consequence of Gödel's incompleteness theorems, HOL with 'standard' semantics is necessarily incomplete. By contrast, automated reasoning in HOL is usually considered with respect to so-called *general models* (introduced by Henkin [35]) in which a meaningful notion of completeness can be achieved (see [1] for a discussion). Broadly speaking, in general models the domain sets $\mathcal{D}_{\alpha\Rightarrow\beta}$ are not necessarily *full*, but still contain enough elements (functions) to guarantee that any term $s_{\alpha\Rightarrow\beta}$ has a denotation. Note that standard models are subsumed under general models, and thus HOL formulas proven valid with respect to the general semantics are also valid in the standard sense. Conversely, any *finite* (counter)model found by a model generator is necessarily standard [3, §54].[27]

---

[27] Current model generators can only find finite (counter)models. Moreover, we shall observe that (classes of only) non-standard models cannot be characterized via HOL formulas [3, §55]. As a consequence, it is not clear how (if at all) the results delivered by HOL automated reasoners

In the context of modal and non-classical logics, a special technique, termed shallow semantical embedding [6,10], has been developed to encode (quantified) non-classical logics into HOL as a meta-language, in such a way that object-logical formulas correspond to HOL terms; this is realized by directly encoding in HOL the truth conditions (semantics) for object-logical connectives as syntactic abbreviations (definitions) in basically the same way as they stand in the textbook[28] This way we can carry out semi-automated[29] object-logical reasoning by translation into a formal meta-logic, HOL, for which good automation support exists. As argued, e.g., in [6], such a shallow embedding allows us to reuse state-of-the-art automated theorem provers and model generators for reasoning with (and about) many different sorts of non-classical logical systems in a very efficient way (avoiding, e.g., inductive definitions and proofs).

The idea of employing HOL as meta-logic to encode *quantified* non-classical logics has been exemplarily discussed by Benzmüller & Paulson in [10] for the case of normal multi-modal logics featuring first-order and propositional quantifiers. Their approach draws upon the 'propositions as sets of worlds' paradigm from modal logic, by adding the twist of encoding sets as their characteristic functions, i.e., as total functions with a (2-valued) Boolean codomain, in such a way that a set S becomes encoded as the function $S(\cdot)$ such that $a \in S$ iff $(S\ a) = \texttt{true}$. In this way, object-logical propositional connectives become easily encoded via the *standard translation* of modal logic into first-order logic. On top of this, Benzmüller & Paulson [10] have shown how to encode object-logical quantifiers (first-order and propositional) by lifting the meta-logical ones. The approach presented in §3.8 builds upon theirs.

3.2   Boolean Algebras

A distinctive feature of our approach consists in encoding Boolean algebras and their extensions by means of their (Stone) representation as algebras of sets (aka. 'fields of sets'). In a nutshell, this means that each element of (the carrier of) the algebra will be a set of what we refer to as *points*. Inspired by the 'propositions as sets of worlds' paradigm from modal logic, we may think of *points* as being 'worlds', and thus of the elements of our Boolean algebras as 'propositions'. Of course, this is just one among many possible interpretations, and nothing stops us from thinking of points as any other kind of object (e.g., they can be sets, functions, sets of functions, etc.). We make use of parameters (Greek letters like ω, α, β) in our types to this end.[30]

---

might ever differ from the 'standard' ones. In the words of Andrews [3, p. 255]: *"one who speaks the language of [HOL] cannot tell whether he lives in a standard or nonstandard world, even if he can answer all the questions he can ask."*

[28] Figuratively speaking, traditional logic texts can be seen as 'embedding' the semantics of object logics into natural language (e.g., English plus mathematical shorthand) as a meta-language.

[29] We employ the expression 'semi-automated' because the process of finding the proofs cannot be fully automated. Proof-checking in HOL is, on the other hand, a much easier problem for which very effective implementations exist. This has motivated the field of *interactive theorem proving* (ITP) which aims at effectively harnessing human ingenuity for proof search, by enabling automated real-time proof-checking feedback via a graphical user interface. Proof assistants (Isabelle, Coq, Lean, etc.) are paradigmatic ITP systems.

[30] Type parameters (e.g., ω, α) are introduced also in our formalization sources [26], using apostrophe-leading symbols (e.g., 'w resp. 'a) which is the notation for type variables in Is-

When given a term of a ω-parameterized type, we take ω as a type whose semantic domain corresponds to the domain or universe of points. For instance, the ω-parametric type $\omega \Rightarrow o$ can be interpreted as the type of (characteristic functions of) sets of points. Given the fundamental role sets (of points) play in this work, we introduce a convenient abbreviation, $(\omega)\sigma$, as (postfix) shorthand notation for the ω-parametric type $\omega \Rightarrow o$. Often, we will simply write σ when the actual type parameter is irrelevant or easily inferred from the context.

We will follow the convention of employing **boldface** for the encoded object-logical algebraic operations ($\wedge, \rightarrow, -, \perp$, etc.), ordering ($\leq$), and equality ($=$). We begin by encoding the latter two as binary predicates on sets, i.e., having type: $\sigma \Rightarrow \sigma \Rightarrow o$.

$$A \leq B \overset{\text{def}}{=} \forall w. (A\ w) \longrightarrow (B\ w)$$

$$A = B \overset{\text{def}}{=} \forall w. (A\ w) \longleftrightarrow (B\ w) \quad (= A \leq B \wedge B \leq A)$$

$$i.e.\ \ A = B = (A\ =\ B)$$

It is worth noting that algebraic equality ($=$) is identical to its meta-logical counterpart ($=_{\sigma \Rightarrow \sigma \Rightarrow o}$) as a consequence of HOL's *functional extensionality* [11]. Observe that the binary operations meet ($\wedge$) and join ($\vee$) become encoded as terms of type $\sigma \Rightarrow \sigma \Rightarrow \sigma$, by reusing HOL's conjunction and disjunction, what makes them tantamount to set intersection and union, respectively. Similarly the top ($\top$) and bottom ($\perp$) elements are encoded as zero-ary connectives (of type σ) by reusing the meta-logical `true` and `false` terms.

$$A \wedge B \overset{\text{def}}{=} \lambda w. (A\ w) \wedge (B\ w) \qquad\qquad \top \overset{\text{def}}{=} \lambda w.\ \texttt{true}$$

$$A \vee B \overset{\text{def}}{=} \lambda w. (A\ w) \vee (B\ w) \qquad\qquad \perp \overset{\text{def}}{=} \lambda w.\ \texttt{false}$$

Drawing upon the classicality of our meta-logic (HOL) we can define an implication connective ($\rightarrow$) analogously by reusing HOL's classical counterpart. Moreover, the object-logical Boolean complement ($-$) can be equivalently encoded by reusing HOL's negation.

$$A \rightarrow B \overset{\text{def}}{=} \lambda w. (A\ w) \longrightarrow (B\ w) \qquad -A \overset{\text{def}}{=} \lambda w. \neg(A\ w)$$

Other related operations, such as difference ($\leftharpoonup$), symmetric difference ($\triangle$), and double implication ($\leftrightarrow$) can be introduced for convenience in the expected way.

We also introduce the infinitary lattice operations $\bigwedge$ and $\bigvee$ which operate on sets of sets (of points) and return a set. Thus, they have as type: $((\omega)\sigma)\sigma \Rightarrow (\omega)\sigma$, explicitly written as: $((\omega \Rightarrow o) \Rightarrow o) \Rightarrow (\omega \Rightarrow o)$.

$$\bigwedge S \overset{\text{def}}{=} \lambda w. \forall X. (S\ X) \longrightarrow (X\ w) \qquad\qquad \bigvee S \overset{\text{def}}{=} \lambda w. \exists X. (S\ X) \wedge (X\ w)$$

Observe that in the spirit of the shallow semantical embedding approach [6,10], $\bigwedge$ and $\bigvee$ reuse the meta-logical quantifiers $\forall$ and $\exists$ (cf. §3.1). It is easy to see (and it has been automatically checked) that they correspond indeed to the lattice-theoretical infimum and supremum operations respectively; i.e. our encoded Boolean algebras are *complete* as lattices.

---

abelle/HOL [43]. This use of type parameters is possible thanks to Isabelle/HOL's support of *rank-1* type polymorphism.

Other convenient notions can become easily encoded. For instance, sets of sets (of points) can be said to be closed under (infinitary) meets or joins.

$$\texttt{meet\_closed } S \overset{\text{def}}{=} \forall X. \forall Y. (S\ X) \wedge (S\ Y) \longrightarrow S\ (X \wedge Y)$$

$$\texttt{join\_closed } S \overset{\text{def}}{=} \forall X. \forall Y. (S\ X) \wedge (S\ Y) \longrightarrow S\ (X \vee Y)$$

$$\texttt{infimum\_closed } S \overset{\text{def}}{=} \forall D.\ D \leq S \longrightarrow (S\ \bigwedge D)$$

$$\texttt{supremum\_closed } S \overset{\text{def}}{=} \forall D.\ D \leq S \longrightarrow (S\ \bigvee D)$$

Note that in the latter two definitions the binary relation $\leq$ is now relating terms of type $((\omega)\sigma)\sigma$.[31] We have also encoded variants in which the subset D is assumed non-empty (which behave slightly differently [26]).

As our goal is to use HOL-reasoning tools for performing object-logical derivations (after unfolding connectives' definitions), some remarks on the 'faithfulness' of our encoding are in order. The first part of the faithfulness question ('soundness') is generally straightforward to resolve by checking that the HOL-encoded axioms and rules of the object logic are (meta-logically) valid. In fact, automated tools can easily be employed to verify this at any time. In the case of the (logic of) Boolean algebras just introduced, this is intuitively evident, since the definitions of the algebraic operations directly reuse HOL's Boolean connectives. The second part of the question ('completeness') involves extra-logical theoretical analysis in order to assess the extent to which the idiosyncratic features of the meta-logic (HOL) have a bearing on the features of the encoded (i.e. modeled) object-logical semantic structures (e.g., equational classes of algebras).[32]

In the present case, we can in fact show that the combination of HOL's λ-abstraction and primitive equality (see §3.1) allows us to synthesize terms that can play the role of Boolean algebra *atoms*. Observe that our encoding validates the following HOL formula:

$$\forall P.\ P \neq \bot \longrightarrow \exists Q.\ \texttt{atom } Q \wedge Q \leq P$$

where $\texttt{atom } A \overset{\text{def}}{=} A \neq \bot \wedge \forall P.\ A \leq P \vee A \leq -P$.

The proof argument is straightforward: just instantiate Q with the term: $\{y\} \overset{\text{def}}{=} (\lambda x.\ x = y)$ for some y such that $(P\ y)$ holds, i.e. Q is the 'singleton' (aka. 'unit-set') generated by y. Hence Boolean algebras, as encoded in our approach, are by default *atomic* (and *complete* as mentioned previously). In fact, under the 'standard' HOL semantics they are just powerset lattices.

In any case, this atomicity restriction does not cause any trouble when it comes to providing semantics for propositional logics based upon (extensions of) Boolean algebras, since, from Stone's representation theorem, we know that every Boolean algebra is a subalgebra of an (atomic) powerset lattice; this entails that if a propositional formula is (counter)satisfiable in a Boolean algebra then it is also (counter)satisfiable in one that is atomic. This way, faithfulness for our HOL-embedded non-classical logics becomes guaranteed at the propositional level in spite of atomicity.

---

[31] Recall that our object-logical terms are type-parameterized, e.g., the relation $\leq$ can be employed for ordering any pair of sets (of sets (of... sets (of points))), i.e., terms of type $(((\omega)\sigma \ldots)\sigma)\sigma$.

[32] Of course, we can simply assume that our object logics are defined semantically via their shallow embeddings (wrt. HOL as a meta-language). This assumption, by default, resolves any questions regarding 'faithfulness' of their encoding.

The question of atomicity becomes actually relevant in the case of propositional and higher-order quantification. In this regard, we note that more general classes of Boolean algebras could, in principle, be obtained in HOL-like systems that do not incorporate extensionality or primitive equality.[33] Another approach could be to introduce specialized quantifiers over sets with appropriate restrictions (excluding singletons, and so forth). We will discuss these types of restrictions in §3.8. In any case, since no clear consensus exists on how quantification should behave in non-classical contexts, we restrict ourselves to providing semantically well-motivated definitions for quantifiers in §3.8. We will not touch upon the issue of faithfulness again in this paper.

3.3 Set-valued Functions and Operators

We will refer to those functions whose codomain consists in sets (of points) as *set-valued functions*. They will be encoded as terms having a (both α- and ω-parametric) type: $\alpha \Rightarrow (\omega)\sigma$ (and we will often employ the Greek letters φ, ψ, η to denote them).

In the spirit of the shallow semantical embedding approach, we can conveniently define the following (α-type-lifted) Boolean operations on set-valued functions.

$$\varphi \ddot{\wedge} \psi \stackrel{\text{def}}{=} \lambda x.\, (\varphi\, x) \wedge (\psi\, x) \qquad\qquad \psi \ddot{\rightarrow} \varphi \stackrel{\text{def}}{=} \lambda x.\, (\psi\, x) \rightarrow (\varphi\, x)$$

$$\varphi \ddot{\vee} \psi \stackrel{\text{def}}{=} \lambda x.\, (\varphi\, x) \vee (\psi\, x) \qquad\qquad \varphi^{-} \stackrel{\text{def}}{=} \lambda x.\, {-}(\varphi\, x)$$

Hence we obtain a *Boolean algebra of set-valued functions*[34] featuring meet ($\ddot{\wedge}$), join ($\ddot{\vee}$) and implication ($\ddot{\rightarrow}$) as binary connectives of type $(\alpha \Rightarrow \sigma) \Rightarrow (\alpha \Rightarrow \sigma) \Rightarrow (\alpha \Rightarrow \sigma)$ and, in particular, *complement* $(\cdot)^{-}$ as a unary connective of type $(\alpha \Rightarrow \sigma) \Rightarrow (\alpha \Rightarrow \sigma)$.

This algebra also features two natural 0-ary connectives (or 'constants') of type $\alpha \Rightarrow \sigma$

$$\ddot{\top} \stackrel{\text{def}}{=} \lambda x.\, \top \quad \text{and} \quad \ddot{\bot} \stackrel{\text{def}}{=} \lambda x.\, \bot\,.$$

Clearly, the usual interrelations between the Boolean algebra connectives obtain, so that this algebra could have also been presented by employing, e.g., the minimal signature $\langle \ddot{\wedge}, (\cdot)^{-} \rangle$. However, we prefer to give a presentation that explicitly reuses the corresponding connectives at the previous level as we did before. We think that this has a pedagogic merit, and, moreover, we have observed that this practice seems to improve the performance of automated reasoning tools.

Analogous to set-valued functions we can also have *set-domain functions*, i.e., terms of type $\sigma \Rightarrow \alpha$. For them we can define a convenient unary operation which we call *dual-complement* having type $(\sigma \Rightarrow \alpha) \Rightarrow (\sigma \Rightarrow \alpha)$, as follows:

$$\varphi^{\overline{\mathsf{d}}} \stackrel{\text{def}}{=} \lambda X.\, \varphi\, ({-}X)\,.$$

---

[33] The presence of primitive equality in HOL and related systems has become commonplace in the wake of the work of Andrews [2]. Automated theorem provers for non-extensional systems are somewhat uncommon, perhaps due to significant limitations (see [2,7] and [8] for a discussion).

[34] Note that set-valued functions are encoded as curried (cf. §3.1) binary relations. Thus, we can easily define (relation) composition and transpose on top of our algebra of set-valued functions to obtain a relation algebra.

We extend the encoding of Boolean algebras by adding further unary operations (with type $\sigma \Rightarrow \sigma$), these we call: *operators*. Moreover, we will often employ an extra (often redundant) pair of parentheses as a suggestive visual aid when applying operators to arguments, so that we will often write $\varphi(A)$ instead of $(\varphi\, A)$. In fact, operators are both set-valued and set-domain functions at the same time. Thus, an *algebra of operators* inherits the connectives defined above plus the following one, which we aptly call *dual*:

$$\varphi^{\mathrm{d}} \overset{\text{def}}{=} \lambda X.\, {-}\varphi(-X)\,.$$

Moreover, given an operator φ, we define the set of its fixed points ($\mathtt{fp}\,\varphi$) as:

$$\mathtt{fp}\,\varphi \overset{\text{def}}{=} \lambda X.\, \varphi(X) = X\,.$$

On top of this we can introduce a further operation $(\cdot)^{\mathrm{fp}}$, dubbed *fixed point*:

$$\varphi^{\mathrm{fp}} \overset{\text{def}}{=} \lambda X.\, \varphi(X) \leftrightarrow X\,,$$

noting that the following fundamental interrelation obtains:

$$(\mathtt{fp}\,\varphi)\, X \longleftrightarrow \varphi^{\mathrm{fp}}(X) = \mathsf{T}\,.$$

We also introduce convenient notation for the complement of the operation $(\cdot)^{\mathrm{fp}}$, which we call *fixed-point-complement*: $\varphi^{\overline{\mathrm{fp}}} \overset{\text{def}}{=} (\varphi^{\mathrm{fp}})^{-} = (\varphi^{-})^{\mathrm{fp}}$. In fact we have that

$$\varphi^{\overline{\mathrm{fp}}} = \lambda X.\, \varphi(X)\ \vartriangle\ X\,.$$

We will often refer to the unary operations (on operators) introduced above, namely $(\cdot)^{-}$, $(\cdot)^{\mathrm{d}}$, $(\cdot)^{\overline{\mathrm{d}}}$, $(\cdot)^{\mathrm{fp}}$ and $(\cdot)^{\overline{\mathrm{fp}}}$, as *transformations*. We note that they are all involutions, i.e., $(\varphi^{f})^{f} = \varphi$ for f in $\{\mathrm{d}, -, \overline{\mathrm{d}}, \mathtt{fp}, \overline{\mathtt{fp}}\}$. We verified several useful interrelations among them, which are best illustrated by means of a commutative diagram, as shown in Fig. 5, which will be instantiated later more concretely as a *topological cube of opposition*.

The *algebra of operators* just introduced can be given in fact a minimal signature: $\langle \boldsymbol{n}, \circ, \ddot{\lambda} \rangle$, where $\boldsymbol{n}$ is a 0-ary connective, defined in the present work as the Boolean complement, i.e., $\boldsymbol{n} \overset{\text{def}}{=} -,$[35] and $\circ$ is an associative binary connective denoting function (operator) composition. We begin by introducing the 0-ary connective $\boldsymbol{e} \overset{\text{def}}{=} \boldsymbol{n} \circ \boldsymbol{n}$, which corresponds in fact to the identity function ($\lambda X.\, X$). Thus $\langle \boldsymbol{e}, \circ \rangle$ is a monoid. We show below how to define the other connectives in terms of $\boldsymbol{n}$, $\boldsymbol{e}$, $\circ$, and $\ddot{\lambda}$:

$$\varphi^{-} \overset{\text{def}}{=} \boldsymbol{n} \circ \varphi$$

$$\varphi^{\overline{\mathrm{d}}} \overset{\text{def}}{=} \varphi \circ \boldsymbol{n}$$

$$\varphi^{\mathrm{d}} \overset{\text{def}}{=} \boldsymbol{n} \circ \varphi \circ \boldsymbol{n}$$

$$\varphi \ddot{\lor} \psi \overset{\text{def}}{=} (\varphi^{-} \ddot{\lambda} \psi^{-})^{-}$$

$$\ddot{\mathsf{T}} \overset{\text{def}}{=} \boldsymbol{n} \ddot{\lor} \boldsymbol{e}$$

$$\ddot{\perp} \overset{\text{def}}{=} \boldsymbol{n} \ddot{\lambda} \boldsymbol{e}$$

$$\varphi^{\mathrm{fp}} \overset{\text{def}}{=} (\varphi \ddot{\lambda} \boldsymbol{e}) \ddot{\lor} (\varphi^{-} \ddot{\lambda} \boldsymbol{n})$$

$$\varphi^{\overline{\mathrm{fp}}} \overset{\text{def}}{=} (\varphi \ddot{\lor} \boldsymbol{e}) \ddot{\lambda} (\varphi^{-} \ddot{\lor} \boldsymbol{n})\,.$$

---

[35] Future work shall explore the consequences of weakening this definition, so that $\boldsymbol{n}$ can stand for an arbitrary involution.
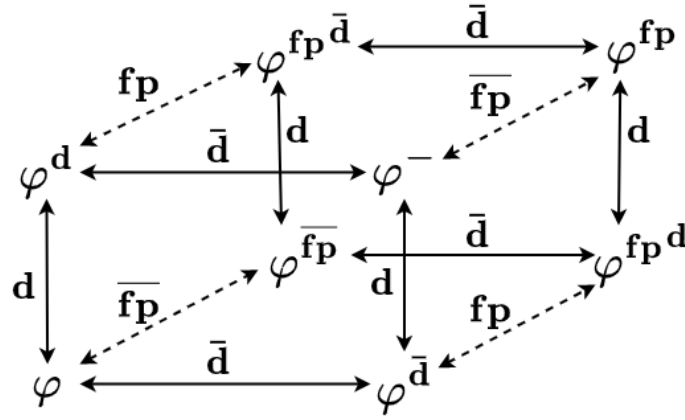
Fig. 5: Topological cube of opposition *in abstracto*. Edges denote (involutive) *transformations* on operators, so that the whole diagram commutes. Note that the diagonals, which correspond to complement $(\cdot)^-$, are not shown to avoid visual clutter.

3.4 Topological Conditions on Operators

Recalling from §2 the axiomatic conditions stated in Def. 1 and Def. 2, corresponding to the Kuratowski closure conditions and their duals, we shall now treat those as properties of unary operations (operators), which we may thus call (for lack of better names): *additive* (ADDI), *multiplicative* (MULT), *expansive* (EXPN), *contractive* (CNTR), *normal* (NORM), *dual-normal* (DNRM), or *idempotent* (IDEM). We shall also suggestively refer to them as a whole as *positive* properties or conditions. We present in Table 3 the corresponding definitions as encoded in HOL. For the sake of finer granularity, definitions involving equalities have been stated separately for the $\leq$ direction and for the $\geq$ direction, conveniently using the suffixes 'a' and 'b' to facilitate reference.[36]

condition (for any φ)	dual condition (for $\psi = \varphi^{\mathrm{d}}$)
ADDI^a $\varphi \overset{\text{def}}{=} \varphi(A \vee B) \leq \varphi(A) \vee \varphi(B)$	MULT^b $\psi \overset{\text{def}}{=} \psi(A \wedge B) \geq \psi(A) \wedge \psi(B)$
ADDI^b $\varphi \overset{\text{def}}{=} \varphi(A \vee B) \geq \varphi(A) \vee \varphi(B)$	MULT^a $\psi \overset{\text{def}}{=} \psi(A \wedge B) \leq \psi(A) \wedge \psi(B)$
EXPN $\varphi \overset{\text{def}}{=} A \leq \varphi(A)$	CNTR $\psi \overset{\text{def}}{=} \psi(A) \leq A$
NORM $\varphi \overset{\text{def}}{=} \varphi(\bot) = \bot$	DNRM $\psi \overset{\text{def}}{=} \psi(\top) = \top$
IDEM^a $\varphi \overset{\text{def}}{=} \varphi(\varphi(A)) \leq \varphi(A)$	IDEM^b $\psi \overset{\text{def}}{=} \psi(A) \leq \psi(\psi(A))$

Table 3: Some *positive* properties of operators (Kuratowski conditions and their duals).

---

[36] Thus ADDI = ADDI^a $\wedge$ ADDI^b, MULT = MULT^a $\wedge$ MULT^b, and IDEM = IDEM^a $\wedge$ IDEM^b. Also observe that IDEM^b and IDEM^a are instances of EXPN and CNTR respectively (therefore we omit them sometimes). Also note that to avoid visual clutter we don't display universal quantifiers in definitions, and thus free variables (A, B, etc.) shall always be read as being universally quantified.

The conditions presented above are not fully independent. We have in fact that the conditions ADDI^b and MULT^a are equivalent. They are actually equivalent to monotonicity (MONO), a fundamental property of operators that deserves special mention:

$$\text{MONO } \varphi \stackrel{\text{def}}{=} A \leq B \longrightarrow \varphi(A) \leq \varphi(B)\,.$$

Thus, we have that $\text{MONO } \varphi$ iff $\text{ADDI}^b \varphi$ iff $\text{MULT}^a \varphi$. In fact, monotonicity is self-dual: $\text{MONO } \varphi$ iff $\text{MONO } \varphi^{\text{d}}$. Clearly, $\text{EXPN } \varphi$ implies both $\text{DNRM } \varphi$ and $\text{IDEM}^b \varphi$, and $\text{CNTR } \varphi$ implies both $\text{NORM } \varphi$ and $\text{IDEM}^a \varphi$.

The infinitary generalizations for ADDI resp. MULT, referred to as iADDI resp. iMULT (cf. Table 4) play an important role in connecting our results to the semantics of normal modal logics, i.e., modal logics characterizable by means of relational (Kripke) frames. We start by noting that, besides entailing ADDI resp. MULT, they also entail NORM resp. DNRM, since the latter can be seen as corresponding to distribution over empty suprema resp. infima.

condition (for any φ)	dual condition (for $\psi = \varphi^{\text{d}}$)
$\text{iADDI}^a \varphi \stackrel{\text{def}}{=} \varphi(\bigvee S) \leq \bigvee [\![\varphi\ S]\!]$	$\text{iMULT}^b \psi \stackrel{\text{def}}{=} \psi(\bigwedge S) \geq \bigwedge [\![\psi\ S]\!]$
$\text{iADDI}^b \varphi \stackrel{\text{def}}{=} \varphi(\bigvee S) \geq \bigvee [\![\varphi\ S]\!]$	$\text{iMULT}^a \psi \stackrel{\text{def}}{=} \psi(\bigwedge S) \leq \bigwedge [\![\psi\ S]\!]$

Table 4: Infinitary *positive* conditions for ADDI (left) and their dual counterparts (right). We employ the notation $[\![\,\cdot\,]\!]$ to denote the image of a set under a function, i.e., $[\![f\ S]\!] \stackrel{\text{def}}{=} \lambda y.\,\exists x.\,(S\ x) \wedge (f\ x) = y$.

In topological terms, closure operators satisfying iADDI (together with the other Kuratowski conditions EXPN and IDEM) are precisely those giving rise to so-called Alexandrov (or 'finitely generated') topologies.[37] It is well known that every Alexandrov topology corresponds, in a sense, to a reflexive and transitive relation (its so-called *specialization preorder*). This correspondence can actually be generalized by factoring out Kuratowski conditions other than iADDI; in fact, the corresponding relation does not need to be a preorder (we may thus refer to it as a *reachability relation*).[38] More specifically,

---

[37] Here we shall remark that iADDI resp. iMULT are strictly stronger conditions than ADDI resp. MULT (e.g., they entail NORM resp. DNRM); they are directly satisfied by closure resp. interior operators defined on relational frames in the usual way (i.e. as the modal operators $\Diamond$ and $\Box$ respectively). Alexandrov spaces can be alternatively characterized as those in which the intersection (union) of an arbitrary family of open (closed) sets is open (closed). In algebraic terms, this means that the set of fixed points of the interior (closure) operator is closed under infinite meets (joins). Of course, both characterizations are equivalent when assuming all Kuratowski conditions. This is, however, not the case for 'weaker' axiomatizations. The interested reader is referred to the formalization sources [26], where we investigate under which *minimal* conditions these results (among others) hold.

[38] It is worth mentioning that in Alexandrov topologies every point has a minimal/smallest neighborhood, namely the set of points related to it by the specialization preorder (or, more generally, the reachability relation).

the following HOL formula is valid:

$$\texttt{iADDI}\ \varphi \longleftrightarrow \varphi = C[\mathcal{R}[\varphi]]\,,$$

where $C[R]$ and $\mathcal{R}[\varphi]$ transform, respectively, a relation into a closure(-like) operator and an operator φ (intended as closure) into a (reachability) relation, i.e.,

$$C[R] \stackrel{\text{def}}{=} \lambda A.\ \lambda w.\ \exists v.\ R\ w\ v \wedge A\ v \quad \text{conversely} \quad \mathcal{R}[\varphi] \stackrel{\text{def}}{=} \lambda w.\ \lambda v.\ \varphi\ \{v\}\ w\,.$$

Let $\psi = C[R]$ for an arbitrary relation R. We have thus that $\texttt{iADDI}\ \psi$ and $\texttt{NORM}\ \psi$ always hold. Moreover, we shall observe that the usual correspondences obtain, as known from modal logic: $\texttt{EXPN}\ \psi$ iff R is reflexive, $\texttt{IDEM}^a\ \psi$ iff R is transitive, and so on. Furthermore, the relation $\mathcal{R}[\varphi]$ will be reflexive if $\texttt{EXPN}\ \varphi$ holds, and transitive if both $\texttt{MONO}\ \varphi$ and $\texttt{IDEM}^a\ \varphi$ hold (see [26]).

In fact we have that the conditions $\texttt{iADDI}^b$ and $\texttt{iMULT}^a$ are equivalent to their finitary versions, and thus $\texttt{MONO}$, $\texttt{ADDI}^b$, $\texttt{MULT}^a$, $\texttt{iADDI}^b$ and $\texttt{iMULT}^a$ are all equivalent. Interesting interrelations with fixed points also obtain: $\texttt{MULT}\ \varphi$ implies $\texttt{meet\_closed(fp}\ \varphi\texttt{)}$ and $\texttt{iMULT}\ \varphi$ implies $\texttt{infimum\_closed(fp}\ \varphi\texttt{)}$, while the converse holds under the additional assumptions $\texttt{MONO}\ \varphi$, $\texttt{CNTR}\ \varphi$ and $\texttt{IDEM}^b\ \varphi$; $\texttt{ADDI}\ \varphi$ implies $\texttt{join\_closed(fp}\ \varphi\texttt{)}$ and $\texttt{iADDI}\ \varphi$ implies $\texttt{supremum\_closed(fp}\ \varphi\texttt{)}$, while the converse holds under the additional assumptions $\texttt{MONO}\ \varphi$, $\texttt{EXPN}\ \varphi$ and $\texttt{IDEM}^a\ \varphi$. Moreover, assuming $\texttt{MONO}\ \varphi$, we have that $\texttt{EXPN}\ \varphi$ implies $\texttt{infimum\_closed(fp}\ \varphi\texttt{)}$ and that $\texttt{CNTR}\ \varphi$ implies $\texttt{supremum\_closed(fp}\ \varphi\texttt{)}$. We refer the interested reader to the Isabelle/HOL sources [26] for more results.

We have just seen how the conditions above can be presented in 'dual pairs', meaning that when an operator φ satisfies one condition, its dual φ^{d} satisfies its dual condition. Another interesting set of 'dual pairs' of conditions is listed in Table 5. These are the conditions satisfied by the *complement* and *dual-complement* of closure operators, to which we suggestively refer as *negative* conditions (whereby the previous ones are their *positive* counterparts).

ref. (any χ)	complement cond. (for $\varphi = \chi^-$)	dual-complement cond. (for $\psi = \varphi^{\text{d}} = \chi^{\bar{\text{d}}}$)
$\texttt{ADDI}^a\ \chi$	$\texttt{nADDI}^a\ \varphi \stackrel{\text{def}}{=} \varphi(A \vee B) \geq \varphi(A) \wedge \varphi(B)$	$\texttt{nMULT}^b\ \psi \stackrel{\text{def}}{=} \psi(A \wedge B) \leq \psi(A) \vee \varphi(B)$
$\texttt{ADDI}^b\ \chi$	$\texttt{nADDI}^b\ \varphi \stackrel{\text{def}}{=} \varphi(A \vee B) \leq \varphi(A) \wedge \varphi(B)$	$\texttt{nMULT}^a\ \psi \stackrel{\text{def}}{=} \psi(A \wedge B) \geq \psi(A) \vee \varphi(B)$
$\texttt{iADDI}^a\ \chi$	$\texttt{inADDI}^a\ \varphi \stackrel{\text{def}}{=} \varphi(\bigvee S) \geq \bigwedge [\![\varphi\ S]\!]$	$\texttt{inMULT}^b\ \psi \stackrel{\text{def}}{=} \psi(\bigwedge S) \leq \bigvee [\![\psi\ S]\!]$
$\texttt{iADDI}^b\ \chi$	$\texttt{inADDI}^b\ \varphi \stackrel{\text{def}}{=} \varphi(\bigvee S) \leq \bigwedge [\![\varphi\ S]\!]$	$\texttt{inMULT}^a\ \psi \stackrel{\text{def}}{=} \psi(\bigwedge S) \geq \bigvee [\![\psi\ S]\!]$
$\texttt{EXPN}\ \chi$	$\texttt{nEXPN}\ \varphi \stackrel{\text{def}}{=} \varphi(A) \leq -A$	$\texttt{nCNTR}\ \psi \stackrel{\text{def}}{=} -A \leq \psi(A)$
$\texttt{NORM}\ \chi$	$\texttt{nNORM}\ \varphi \stackrel{\text{def}}{=} \varphi(\bot) = \top$	$\texttt{nDNRM}\ \psi \stackrel{\text{def}}{=} \psi(\top) = \bot$
$\texttt{IDEM}^a\ \chi$	$\texttt{nIDEM}^a\ \varphi \stackrel{\text{def}}{=} \varphi(A) \leq \varphi(-\varphi(A))$	$\texttt{nIDEM}^b\ \psi \stackrel{\text{def}}{=} \psi(-\psi(A)) \leq \psi(A)$

Table 5: Some *negative* conditions (taking closure conditions as reference).

The prefix 'n' in the names of the conditions in Table 5 can suggestively be read as '*anti*', and thus, e.g., $\texttt{nADDI}$, $\texttt{nEXPN}$ and $\texttt{nIDEM}$ could be read as *anti-additivity*, *anti-expansiveness* and *anti-idempotence*, respectively. We can also define the negative counterpart to the monotonicity condition, dubbed *antitonicity* or *anti-monotonicity* ($\texttt{ANTI}$):

$$\texttt{ANTI}\ \varphi \stackrel{\text{def}}{=} A \leq B \longrightarrow \varphi(B) \leq \varphi(A)\,.$$

Unsurprisingly, `ANTI` is self-dual. In fact, analogously as before, `ANTI`, `nADDI`^{b}, `nMULT`^{a}, `inADDI`^{b}, and `inMULT`^{a} end up being all equivalent. Analogous interrelations obtain here as in the previous case, replacing the conditions by their complement (negative) counterparts (see sources [26]).

In fact, the interrelations between the axiomatic conditions satisfied by an operator φ, its dual φ^d, its complement φ^- and its dual-complement $\varphi^{\bar{d}}$ (Tables 3, 4 and 5), can also be conveniently represented in a kind of *square of opposition*, as illustrated in Fig. 6.
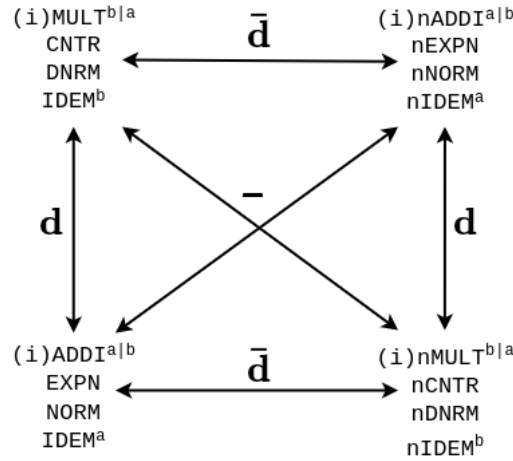

Fig. 6: Interrelations between some positive and negative conditions for operators.

In the same spirit, we shall now introduce pairs of related conditions corresponding to the fixed-point transformations $(\cdot)^{\mathtt{fp}}$ and $(\cdot)^{\overline{\mathtt{fp}}}$. However, our computer-supported experiments [26] quickly led us to realize that the axiomatic conditions discussed so far are too coarse-grained to allow for establishing one-to-one correspondences (as in Tables 3-5). In fact, the axiomatic conditions need to be considered in what we call a 'relativized' form.

Broadly speaking, relativizing an axiomatic condition, in our sense, consists in restricting the domain of points (in equalities or operators) wrt. a 'dynamically' constructed set. This 'relativizing' set is thus not fixed but defined in terms of the (universally quantified) variables featured in the definition. As an illustration, the condition

$$\mathtt{nMULT}\ \psi \stackrel{\mathrm{def}}{=} \psi(A \wedge B)\ = \psi(A) \vee \psi(B)\,,$$

which is equivalently stated (with explicit quantification over points) as

$$\mathtt{nMULT}\ \psi \stackrel{\mathrm{def}}{=} \forall w.\ \psi(A \wedge B)\ w \longleftrightarrow (\psi(A) \vee \psi(B))\ w\,,$$

can be relativized as

$$\mathtt{nMULTr}\ \psi \stackrel{\mathrm{def}}{=} \forall w.\ (A \wedge B)\ w \longrightarrow (\psi(A \wedge B)\ w \longleftrightarrow (\psi(A) \vee \psi(B))\ w)\,,$$

which is, in fact, equivalently formulated as

$$\texttt{nMULTr}\ \psi \overset{\text{def}}{=} (A \wedge B) \wedge \psi(A \wedge B) = (A \wedge B) \wedge (\psi(A) \vee \psi(B)).$$

We take the opportunity to observe that $\texttt{nMULTr}$ above corresponds to the axiomatic condition B1 for topological *border* operators, as presented in Def. 4 in §2. This is of course not a coincidence, as will be discussed in §3.5.

We shall now introduce a convenient notation for (in)equalities relativized wrt. the points inside resp. outside of a given set U

$$A =_U B \overset{\text{def}}{=} \forall x.\ U\,x \longrightarrow (A\,x \longleftrightarrow B\,x) \qquad A \leq_U B \quad \overset{\text{def}}{=} \forall x.\ U\,x \longrightarrow (A\,x \longrightarrow B\,x)$$

$$A =^U B \overset{\text{def}}{=} \forall x.\ \neg U\,x \longrightarrow (A\,x \longleftrightarrow B\,x) \qquad A \leq^U B \quad \overset{\text{def}}{=} \forall x.\ \neg U\,x \longrightarrow (A\,x \longrightarrow B\,x)$$

which we shall call the *lower* resp. *upper* relativization (of a given (in)equality) wrt. an element U (in our case a set). Observe that the definitions above have alternative equivalent formulations as

$$A =_U B = (U \wedge A = U \wedge B) \qquad\qquad A \leq_U B = (U \wedge A \leq U \wedge B)$$

$$A =^U B = (U \vee A = U \vee B) \qquad\qquad A \leq^U B = (U \vee A \leq U \vee B).$$

These definitions above can thus be interpreted as the relativization of (in)equalities wrt. the elements below resp. above U; or, algebraically speaking, wrt. the principal ideal resp. filter generated by U.

We now list the corresponding relativized variants for axiomatic conditions. We begin by noting that, in our case, the relativizing terms can at most feature only *bound* variables (see e.g. $(A \wedge B)$ in $\texttt{nMULTr}$ above). As a consequence, relativization cannot be meaningfully applied to conditions $(\texttt{n})\texttt{NORM}$ or $(\texttt{n})\texttt{DNRM}$. Similarly, relativizing the conditions $(\texttt{n})\texttt{EXPN}$ or $(\texttt{n})\texttt{CNTR}$ either leaves them unchanged or trivializes them. Thus, relativization meaningfully applies only to additivity and idempotence and derived conditions. We show in Tables 6 and 7 the *appropriate* (for our purposes) relativized variants of the positive and negative axiomatic conditions previously discussed.[39]

relativized condition (for any φ)	relativized dual condition (for $\psi = \varphi^{\text{d}}$)
$\texttt{ADDIr}^a\ \varphi \overset{\text{def}}{=} \varphi(A \vee B) \leq^{A \vee B} \varphi(A) \vee \varphi(B)$	$\texttt{MULTr}^b\ \psi \overset{\text{def}}{=} \psi(A \wedge B) \geq_{A \wedge B} \psi(A) \wedge \psi(B)$
$\texttt{ADDIr}^b\ \varphi \overset{\text{def}}{=} \varphi(A \vee B) \geq^{A \vee B} \varphi(A) \vee \varphi(B)$	$\texttt{MULTr}^a\ \psi \overset{\text{def}}{=} \psi(A \wedge B) \leq_{A \wedge B} \psi(A) \wedge \psi(B)$
$\texttt{iADDIr}^a\ \varphi \overset{\text{def}}{=} \varphi(\bigvee S) \leq^{\bigvee S} \bigvee[\![\varphi\ S]\!]$	$\texttt{iMULTr}^b\ \psi \overset{\text{def}}{=} \psi(\bigwedge S) \geq_{\bigwedge S} \bigwedge[\![\psi\ S]\!]$
$\texttt{iADDIr}^b\ \varphi \overset{\text{def}}{=} \varphi(\bigvee S) \geq^{\bigvee S} \bigvee[\![\varphi\ S]\!]$	$\texttt{iMULTr}^a\ \psi \overset{\text{def}}{=} \psi(\bigwedge S) \leq_{\bigwedge S} \bigwedge[\![\psi\ S]\!]$
$\texttt{IDEMr}^a\ \varphi \overset{\text{def}}{=} \varphi(A \vee \varphi(A)) \leq^A \varphi(A)$	$\texttt{IDEMr}^b\ \psi \overset{\text{def}}{=} \psi(A) \leq_A \psi(A \wedge \psi(A))$

Table 6: Relativized variants for some *positive* properties of operators (cf. Tables 3-4).

---

[39] Observe that when iterating operations (as in $(\texttt{n})\texttt{IDEM}$) the relativization of their domain becomes evident. These relativized variants have been elicited with the crucial help of automated tools, in particular Isabelle's *Sledgehammer* [12] and *Nitpick* [13].

relativized condition (for any φ)	relativized dual condition (for $\psi = \varphi^{\mathrm{d}}$)
$\texttt{nADDIr}^a\ \varphi \overset{\text{def}}{=} \varphi(A \vee B) \geq^{A \vee B} \varphi(A) \wedge \varphi(B)$	$\texttt{nMULT}^b\ \psi \overset{\text{def}}{=} \psi(A \wedge B) \leq_{A \wedge B} \psi(A) \vee \varphi(B)$
$\texttt{nADDIr}^b\ \varphi \overset{\text{def}}{=} \varphi(A \vee B) \leq^{A \vee B} \varphi(A) \wedge \varphi(B)$	$\texttt{nMULT}^a\ \psi \overset{\text{def}}{=} \psi(A \wedge B) \geq_{A \wedge B} \psi(A) \vee \psi(B)$
$\texttt{inADDIr}^a\ \varphi \overset{\text{def}}{=} \varphi(\bigvee S) \geq^{\vee S} \bigwedge [\![\varphi\ S]\!]$	$\texttt{inMULTr}^b\ \psi \overset{\text{def}}{=} \psi(\bigwedge S) \leq_{\wedge S} \bigvee [\![\psi\ S]\!]$
$\texttt{inADDIr}^b\ \varphi \overset{\text{def}}{=} \varphi(\bigvee S) \leq^{\vee S} \bigwedge [\![\varphi\ S]\!]$	$\texttt{inMULTr}^a\ \psi \overset{\text{def}}{=} \psi(\bigwedge S) \geq_{\wedge S} \bigvee [\![\psi\ S]\!]$
$\texttt{nIDEMr}^a\ \varphi \overset{\text{def}}{=} \varphi(A) \leq^A \varphi(A \vee -\varphi(A))$	$\texttt{nIDEMr}^b\ \psi \overset{\text{def}}{=} \psi(A \wedge -\psi(A)) \leq_A \psi(A)$

Table 7: Relativized variants for some *negative* properties of operators (cf. Table 5).

It is worth to have a look at the two pairs of dual conditions below, corresponding to weak variants of MONO and ANTI.

$$\texttt{MONOw}^1\ \varphi \overset{\text{def}}{=} A \leq B \longrightarrow \varphi(A) \leq B \vee \varphi(B)$$

$$\texttt{MONOw}^2\ \varphi \overset{\text{def}}{=} A \leq B \longrightarrow A \wedge \varphi(A) \leq \varphi(B)$$

$$\texttt{ANTIw}^1\ \varphi \overset{\text{def}}{=} A \leq B \longrightarrow \varphi(B) \leq B \vee \varphi(A)$$

$$\texttt{ANTIw}^2\ \varphi \overset{\text{def}}{=} A \leq B \longrightarrow A \wedge \varphi(B) \leq \varphi(A)$$

Unsurprisingly, $\texttt{MONOw}^1$ and $\texttt{MONOw}^2$ are each other's duals. The same holds for $\texttt{ANTIw}^1$ and $\texttt{ANTIw}^2$. In fact, analogously as before, $\texttt{MONOw}^1$ is equivalent to $\texttt{ADDIr}^b$ and $\texttt{MONOw}^2$ is equivalent to $\texttt{MULTr}^a$. Similarly, $\texttt{ANTIw}^1$ is equivalent to $\texttt{nADDIr}^b$ and $\texttt{ANTIw}^2$ is equivalent to $\texttt{nMULTr}^a$ (see sources [26] for more interrelations).

The conditions (n)EXPN and (n)CNTR play a fundamental role in relating axiomatic conditions with their relativized variants. Thus,

- given any of EXPN or nEXPN, we have that $\texttt{ADDI}^a$ and $\texttt{ADDIr}^a$ are equivalent;
- given EXPN, we have that $\texttt{ADDI}^b$ and $\texttt{ADDIr}^b$ (and $\texttt{MONOw}^1$) are equivalent;
- given CNTR, we have that $\texttt{MULT}^a$ and $\texttt{MULTr}^a$ (and $\texttt{MONOw}^2$) are equivalent;
- given any of CNTR or nCNTR, we have that $\texttt{MULT}^b$ and $\texttt{MULTr}^b$ are equivalent;
- given any of EXPN or nEXPN, we have that $\texttt{nADDI}^a$ and $\texttt{nADDIr}^a$ are equivalent;
- given nEXPN, we have that $\texttt{nADDI}^b$ and $\texttt{nADDIr}^b$ (and $\texttt{ANTIw}^1$) are equivalent;
- given nCNTR, we have that $\texttt{nMULT}^a$ and $\texttt{nMULTr}^a$ (and $\texttt{ANTIw}^2$) are equivalent;
- given any of CNTR or nCNTR, we have that $\texttt{nMULT}^b$ and $\texttt{nMULTr}^b$ are equivalent.

Having discussed relativized axiomatic conditions and their interrelations, we can now return to the question of establishing one-to-one correspondences between conditions and their fixed-point transformations. First, observe that

$$\texttt{EXPN}\ \varphi \quad \text{iff} \quad \texttt{EXPN}\ \varphi^{\mathrm{fp}} \quad \text{iff} \quad \texttt{nEXPN}\ \varphi^{\overline{\mathrm{fp}}}$$

$$\texttt{CNTR}\ \varphi \quad \text{iff} \quad \texttt{nCNTR}\ \varphi^{\mathrm{fp}} \quad \text{iff} \quad \texttt{CNTR}\ \varphi^{\overline{\mathrm{fp}}}$$

and also

$$\texttt{NORM}\ \varphi \quad \text{iff} \quad \texttt{nNORM}\ \varphi^{\mathrm{fp}} \quad \text{iff} \quad \texttt{NORM}\ \varphi^{\overline{\mathrm{fp}}}$$

$$\texttt{DNRM}\ \varphi \quad \text{iff} \quad \texttt{DNRM}\ \varphi^{\mathrm{fp}} \quad \text{iff} \quad \texttt{nDNRM}\ \varphi^{\overline{\mathrm{fp}}}.$$

As for relativized variants of additivity and derived negative conditions we have that

$$\text{(i)ADDIr}^a\,\varphi \quad \text{iff} \quad \text{(i)nADDIr}^a\,\varphi^{\text{fp}} \quad \text{iff} \quad \text{(i)ADDIr}^a\,\varphi^{\overline{\text{fp}}}$$

$$\text{(i)ADDIr}^b\,\varphi \quad \text{iff} \quad \text{(i)nADDIr}^b\,\varphi^{\text{fp}} \quad \text{iff} \quad \text{(i)ADDIr}^b\,\varphi^{\overline{\text{fp}}}$$

and similarly for multiplicativity and derived negative conditions

$$\text{(i)MULTr}^a\,\varphi \quad \text{iff} \quad \text{(i)MULTr}^a\,\varphi^{\text{fp}} \quad \text{iff} \quad \text{(i)nMULTr}^a\,\varphi^{\overline{\text{fp}}}$$

$$\text{(i)MULTr}^b\,\varphi \quad \text{iff} \quad \text{(i)MULTr}^b\,\varphi^{\text{fp}} \quad \text{iff} \quad \text{(i)nMULTr}^b\,\varphi^{\overline{\text{fp}}}.$$

Finally, in the case of IDEM and derived negative conditions we have that

$$\text{IDEMr}^a\,\varphi \quad \text{iff} \quad \text{nIDEMr}^a\,\varphi^{\text{fp}} \quad \text{iff} \quad \text{IDEMr}^a\,\varphi^{\overline{\text{fp}}}$$

$$\text{IDEMr}^b\,\varphi \quad \text{iff} \quad \text{IDEMr}^b\,\varphi^{\text{fp}} \quad \text{iff} \quad \text{nIDEMr}^b\,\varphi^{\overline{\text{fp}}}.$$

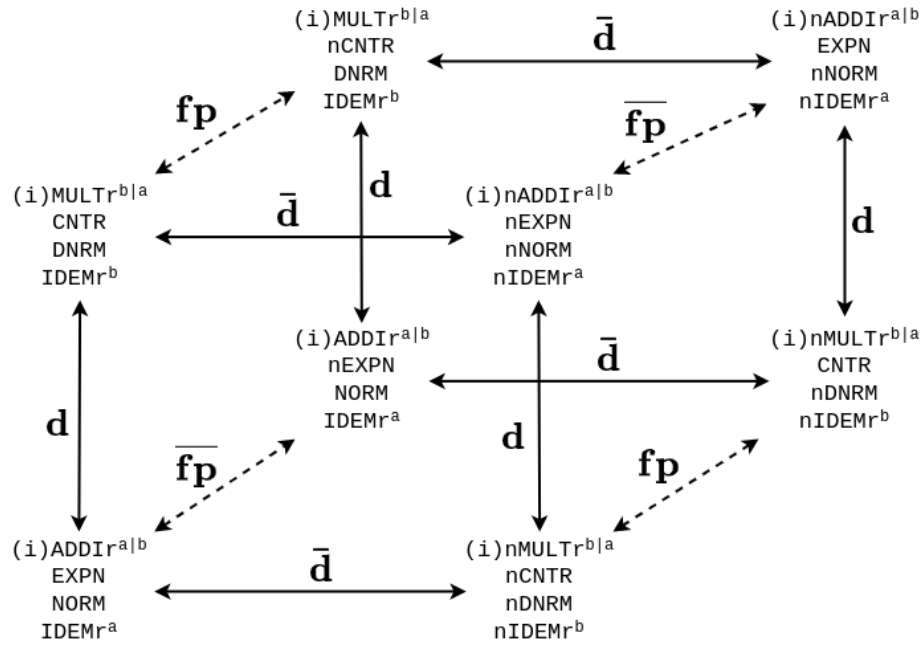In Fig. 7 we show a suggestive representation of the interrelations between the axiomatic conditions discussed so far.

Fig. 7: Interrelations between some axiomatic conditions for operators. The lower left corner corresponds to the Kuratowski closure conditions.

3.5   A Topological Cube of Opposition

We have previously remarked that the four conditions: ADDI, EXPN, NORM and IDEM, correspond to the well-known Kuratowski conditions on *closure* operators (recall §2 Def. 1), and that any operator C satisfying some 'meaningful' subset of those (possibly weakened) conditions shall be called a *closure operator*.[40] Similarly, any operator I partially satisfying the dual conditions (MULT, CNTR, DNRM, IDEM) shall be called an *interior operator* (cf. §2 Def. 2). Recall, moreover, that we always obtain an interior operator as the dual of a closure: $C^d = I$. We recall from §2 that operators in TBAs are inter-definable (see Table 8 and Fig. 8).

define\using	C	I	E	B
$C(A)$	–	$I^d(A)$	$-E(A)$	$A \vee B(-A)$
$I(A)$	$C^d(A)$	–	$E(-A)$	$A \wedge -B(A)$
$E(A)$	$-C(A)$	$I(-A)$	–	$-A \wedge B^d(A)$
$B(A)$	$A \wedge C(-A)$	$A \wedge -I(A)$	$A \wedge E^d(A)$	–

Table 8: Inter-definitions between operators (extract from Table 2). Recall that entries involving border (B) require assuming the corresponding second Kuratowski condition (i.e., EXPN for C, CNTR for I, and so on; cf. §2.3).

The reader may have noticed that the negative conditions for $(\cdot)^-$ presented in Table 5 correspond indeed to the conditions axiomatizing an exterior operator (cf.§2 Def. 3). Of course, this is not surprising if we recall the inter-definitions between topological operators in Table 8.

Let us now instantiate the commutative diagram shown in Fig. 5 with a closure operator C, and notice that its frontal face corresponds to the *square of opposition* shown in Fig. 9, which constitutes itself the frontal face of what we suggestively refer to as a *topological cube of opposition* later in Fig. 11.

Recalling the axiomatic conditions on a *border* operator B (see §2 Def. 4) the reader may also notice, by looking at Fig. 7, that the conditions satisfied by $\left((\cdot)^{\bar{d}}\right)^{fp} (= ((\cdot)^{fp})^d)$ for a closure operator closely resemble (and are in fact equivalent to[41]) those of a border operator. This is not a coincidence. In fact a border operator B can be defined as $(C^{fp})^d$ for a given closure operator C. Thus, we can notice that the rear face of the commutative diagram in Fig. 5 (instantiated with a closure operator C) gives rise to the *square of opposition* shown in Fig. 10.

---

[40] As noted before, we are abusing terminology, as we do not fix a minimal set of conditions that an operator needs to satisfy to deserve being called a 'closure'; e.g., Moore (hull) closures only satisfy MONO, EXPN, and IDEM, while Čech closure operators may not satisfy IDEM (they are sometimes called 'preclosures'). Anyhow, we let the context dictate how operators are called.

[41] Observe that the first three conditions coincide. As for the last condition, it can be shown that, in the presence of the other conditions, B4 is in fact equivalent to nIDEMr^b (see [26]).
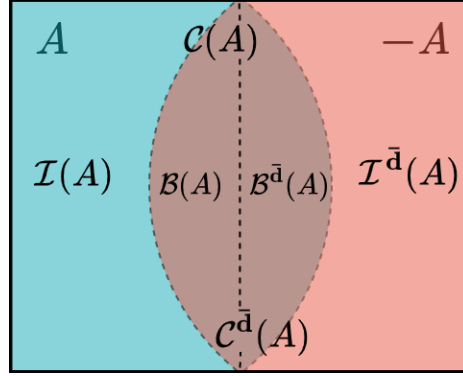
Fig. 8: Suggestive pictorial representation of topological operators and their inter-definitions. Note that the illustration implicitly assumes the second Kuratowski condition (i.e., EXPN for C and its counterpart for the other operators).
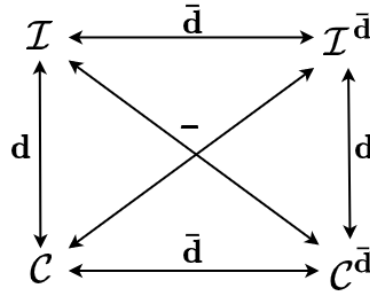

Fig. 9: Frontal face of the *topological cube of opposition* (instantiated with C).
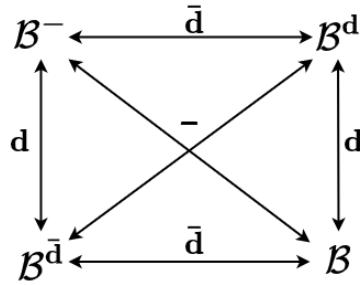

Fig. 10: Rear face of the *topological cube of opposition* (instantiated with $\mathcal{B} = (C^{\mathrm{fp}})^{\mathrm{d}}$).

Let us now assemble our two squares in Fig. 9 and Fig. 10 into the *topological cube of opposition* in Fig. 11. It is instructive to compare the latter with the corresponding axiomatic conditions in Fig. 7.
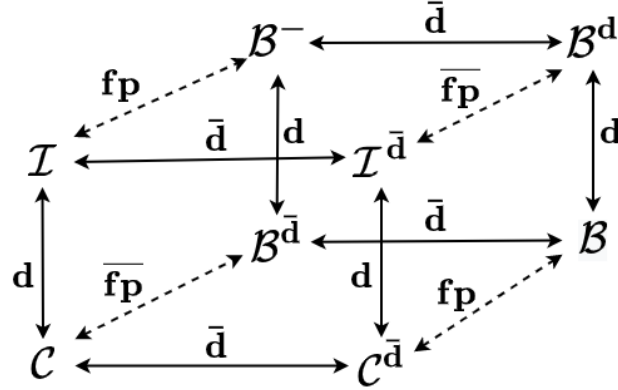

Fig. 11: *Topological cube of opposition* instantiated with C (and $B = (C^{\mathbf{fp}})^{\mathbf{d}}$). Note that, for the sake of readability, the diagonals are not shown. They can be easily inferred.

It is also a good exercise to contemplate the cube in Fig. 11 from different perspectives (from above/below, laterally, diagonally, etc.) as it gives rise to quite interesting commutative diagrams. Also interesting is to employ the operations $\ddot{\curlywedge}$ and $\ddot{\curlyvee}$ to explore symmetries and dualities between pairs of operators, e.g., at opposed corners. As an illustration, we can contemplate the 'floor' square and recall that a frontier operator $\mathcal{F}$ is equivalently defined as $C \ddot{\curlywedge} C^{\overline{\mathbf{d}}}$ and $B^{\overline{\mathbf{d}}} \ddot{\curlyvee} B$ (cf. §2).

3.6 Non-classical Negations

It is well known that closure algebras can serve as semantic structures for modal logic, interpreting the diamond as a closure operator C. By enforcing all the Kuratowski closure conditions on C, this semantics becomes complete for modal logic **S4**. Naturally, this result can be generalized towards weaker closure(-like) operators. For instance, Jónsson and Tarski [37] demonstrated that by constraining C with only the conditions ADDI and NORM we obtain a complete semantics for the minimal normal modal logic **K** (as we call it nowadays). Therefore, it is reasonable to think of the (suitably constrained) operator C and its dual $C^{\mathbf{d}} = \mathcal{I}$ as *positive modalities*.

On the other hand, we have just seen that other interesting operators are definable from C by means of the transformations $(\cdot)^-, (\cdot)^{\mathbf{d}}, (\cdot)^{\overline{\mathbf{d}}}, (\cdot)^{\mathbf{fp}}$ and $(\cdot)^{\overline{\mathbf{fp}}}$ as well as the connectives: $\ddot{\curlywedge}, \ddot{\curlyvee}$ and $\dot{\rightarrow}$. Moreover operators can also be composed as functions. In particular, recalling Fig. 8, we can see that the operators $C^{\overline{\mathbf{d}}} = \mathcal{I}^-$ and $C^- = \mathcal{I}^{\overline{\mathbf{d}}} = \mathcal{E}$ make good candidates for paraconsistent and paracomplete negations (*negative modal-*

ities), respectively. Let us define: $\neg^C \overset{\text{def}}{=} C^{\bar{\text{d}}}$ and $\neg^I \overset{\text{def}}{=} I^{\bar{\text{d}}} = (\neg^C)^{\text{d}}$.[42] By employing *degree-preserving consequence* (recall §2.4) it is easy to see that 'explosion' (ECQ) resp. 'excluded middle' (TND) do not hold for $\neg^C$ resp. $\neg^I$, i.e.,

$$A, \neg^C A \nvdash \bot \quad \text{and} \quad \nvdash A, \neg^I A.$$

Certainly, for $\neg^C$ or $\neg^I$ to behave as a negation worthy of its name, certain conditions must be met by the underlying operator C.[43] The characterization of a 'bona fide negation' is a contentious philosophical topic that is orthogonal to the present discussion. We limit ourselves to noting that results such as those presented in §3.4 allow us to 'translate' conditions on C into their counterparts for other operators (in particular $C^- = \neg^I$ and $C^{\bar{\text{d}}} = \neg^C$). This essentially provides a mechanism to control the behavior of potential negations (see, e.g., the properties in Table 1) by suitably constraining a primitive closure(-like) operator. The interested reader can find in our Isabelle/HOL sources [26] a (necessarily partial but steadily growing) list of results in this regard.

It is worth quoting at this point the *square of modalities* as discussed by Marcos [40] (and shown in Fig. 12, left), since it will help us set the stage for the discussion on recovery operators in §3.7. It is worth mentioning that Marcos [40] has argued (rightly in our opinion) for a more appropriate interpretation of the vertical edges of the square as "dualitas" instead of "subalternatio". Observe that this coincides with our previous depiction of the *topological cube of opposition* (see Fig. 12, right).
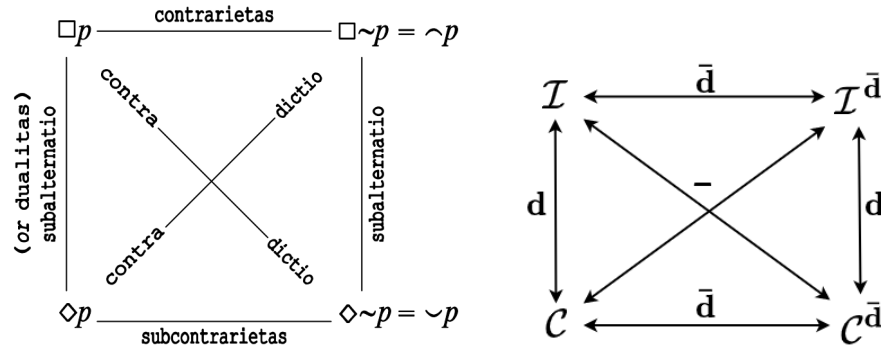

Fig. 12: On the left: Marcos' *square of modalities* (excerpt from [40]). The connectives $\sim$, $\cap$ and $\smile$ correspond to classical, paracomplete and paraconsistent negation respectively. On the right: front face of the *topological cube of opposition* (as in Fig. 9).

---

[42] The idea of defining non-classical negations by composing a classical negation with a modality is of course not new and has been proposed in countless occasions by researchers working at the crossroads of modal and non-classical logics (see e.g. [40,22] and references therein).

[43] As one reviewer noted, similar definitions for non-classical negations and related modalities in the literature (e.g., [40,22,21] and references therein) hinge on the validity of EXPN for C (and its counterparts for other operators: CNTR for I, nEXPN for $\mathcal{E}$, etc.).

Finally, we can speculate on the existence of other interesting negation-like connectives obtained by composing operators with each other. Consider, for example, the negation(-like) operators: $\neg^{IC} \overset{\text{def}}{=} I \circ C^{\overline{\text{d}}}$ and $\neg^{CI} \overset{\text{def}}{=} C \circ I^{\overline{\text{d}}}$ for operators C and I (either both being primitive or one defined in terms of the other). Preliminary computer-assisted experiments [26] have shown that, by imposing suitable axiomatic (e.g, Kuratowski) conditions on the operators, we can obtain several potentially useful kinds of paradefinite logics in this way. Generally speaking, operations defined according to the schema: $\neg^X \overset{\text{def}}{=} \lambda A.\ g(C, -)(A)$, where $g(C, -)$ is an arbitrary composition of C and $-$, with the latter appearing an odd number of times, are expected to behave as negation(-like) operators. Recalling the limiting result by Kuratowski mentioned in §2, it follows that we can define at most seven different negations in this manner, restricting ourselves to only one primitive operator (either C or I). However, keep in mind that Kuratowski's result was obtained for fully constrained (closure) operators. We can therefore anticipate this number to increase as we relax those restrictions. Another viable alternative involves working with several suitably interrelated *primitive* operators (as in the *bi-topological Boolean algebras* studied by Rauszer [45]). Given the multitude of ways these extensions can be performed, we consider this a potentially interesting topic for future computer-assisted investigations.

3.7   Recovery Operators

We now shift our focus to the notion of a fixed point as previously discussed. We begin by emphasizing that the fixed-point predicate $(\texttt{fp}\ \cdot)$ provides a mechanism for 'recovering' many of the classical properties of negation in a sentence-wise fashion.

Regarding the paraconsistent case, we have observed that $\neg^C$ does not validate ECQ as anticipated. As it happens, we can 'switch on' ECQ for a particular A by simply assuming that A is open, i.e., $I(A) = A$ or in other words $(\texttt{fp}\ I)\ A$. Hence, we have

$$(\texttt{fp}\ I)\ A \longrightarrow A \wedge \neg^C A = \bot \quad \text{i.e. if A is open then } A, \neg^C A \vdash \bot.$$

Dually, when using the paracomplete negation $\neg^I$, TND can be recovered for a given A by assuming that A is closed. Other 'negation-like' semantic conditions (e.g., as in Table 1) can be similarly recovered by assuming some *minimal* conditions on C. These results have been obtained using Isabelle/HOL; therefore, we direct interested readers to [26] for a list of these findings. Furthermore, these results can be easily given an intuitive topological explanation if we recall from §2 that fixed points of operators have interesting alternative characterizations in terms of algebras of sets (e.g., open sets are those with an empty border, and closed sets are those whose complement has an empty border).

The connection to *object-logical* recovery operators can be established by recalling from §3.3 that the fixed-point predicate $(\texttt{fp}\ \cdot)$ can be 'operationalized' into $(\cdot)^{\texttt{fp}}$, such that the following holds:

$$(\texttt{fp}\ \varphi)\ A \longleftrightarrow \varphi^{\texttt{fp}}(A) = \top.$$

This entails in fact that any 'recovery results' obtained by assuming $(\mathtt{fp}\ \varphi)$, as discussed above, extend directly to $\varphi^{\mathtt{fp}}$. In fact, the following holds in HOL:[44]

$$\varphi^{\mathtt{fp}}(A) \wedge (\Gamma\ A) \leq (\Delta\ A) \longrightarrow ((\mathtt{fp}\ \varphi)\ A \longrightarrow (\Gamma\ A) \leq (\Delta\ A)).$$

As for the paraconsistent case, we note that ECQ (among others) can be recovered by purely object-logical (algebraic) means. Let us introduce the connective $\circ \overset{\text{def}}{=} \mathcal{I}^{\mathtt{fp}}$ (= $\mathcal{B}^-$), which we shall call a *consistency* operator, inspired by the Logics of Formal Inconsistency (LFIs) [18,15,17]. From the previous discussion it is easy to see that the LFIs' *principle of gentle explosion* obtains:

$$\circ A \wedge A \wedge \neg^C A = \bot \quad \text{i.e.}\ \circ A, A, \neg^C A \vdash \bot.$$

Of course, similar results hold in the dual, paracomplete setting. Let us introduce the connective $\star \overset{\text{def}}{=} C^{\mathtt{fp}}$ (= $\mathcal{B}^{\mathtt{d}}$), which we call a *determinedness* operator, inspired by the Logics of Formal Undeterminedness (LFUs) [40,17]. We can see that the following holds:

$$\star A \leq A \vee \neg^{\mathcal{I}} A \quad \text{i.e.}\ \vdash \bigstar A, A, \neg^{\mathcal{I}} A \quad (\text{for } \bigstar A \overset{\text{def}}{=} -\star A).$$

We shall remark that the results stated above (in terms of operators C and $\mathcal{I}$) have been obtained without assuming any particular restrictions (e.g., Kuratowski conditions). This motivates a more general characterization of recovery operators which will be discussed below. In the meanwhile, before we move the discussion to a more abstract level, it is interesting to recall the *square of perfections* for recovery operators ($\circ$ resp. $\star$ and their complements $\bullet$ resp. $\bigstar$) as introduced by Marcos [40] and reproduced slightly amended[45] in Fig. 13, where it is compared to our *topological cube of opposition*.

We now characterize recovery operators in a more abstract fashion. First note that, for any arbitrary operator φ, if we were to treat $\varphi^{\overline{\mathtt{d}}}$ as a negation, then we can see that both ECQ and TND are valid for the fixed points of φ's dual:[46]

$$(\mathtt{fp}\ \varphi^{\mathtt{d}})\ A \longrightarrow A \wedge \varphi^{\overline{\mathtt{d}}}(A) = \bot \quad \text{dually} \quad (\mathtt{fp}\ \varphi^{\mathtt{d}})\ A \longrightarrow A \vee \varphi^{\overline{\mathtt{d}}}(A) = \top.$$

Hence, we can obtain a generalized *principle of gentle explosion* and its dual:

$$(\varphi^{\mathtt{d}})^{\mathtt{fp}}(A) \wedge A \wedge \varphi^{\overline{\mathtt{d}}}(A) = \bot \quad \text{dually} \quad (\varphi^{\mathtt{d}})^{\mathtt{fp}}(A) \leq A \vee \varphi^{\overline{\mathtt{d}}}(A),$$

$$(\varphi^{\mathtt{d}})^{\mathtt{fp}}(A), A, \varphi^{\overline{\mathtt{d}}}(A) \vdash \bot \quad \text{dually} \quad \vdash (\varphi^{\overline{\mathtt{d}}})^{\mathtt{fp}}(A), A, \varphi^{\overline{\mathtt{d}}}(A).$$

---

[44] Observe that the terms Γ and Δ, having type $\sigma \Rightarrow \sigma$, can be seen intuitively as corresponding to (arbitrary) formulas with a free variable A (of type σ). In fact, this HOL-statement can be seamlessly extended to any number of variables.

[45] In [40] Marcos refers to $\star$ as a *determinedness* operator (as we do) but gives it a characterization that is more appropriate for modelling *undeterminedness* (for which we employ $\bigstar$). This issue has also been highlighted by [17]. As a consequence, the positions for $\star$ and $\bigstar$ have been swapped in Fig. 13 in contrast to the original diagram as presented by Marcos in [40].

[46] To see why, the reader can first notice that the set of fixed points of $\varphi^{\mathtt{d}}$ is identical to the set of complements of the fixed points of φ, i.e., $(\mathtt{fp}\ \varphi^{\mathtt{d}}) = (\mathtt{fp}\ \varphi)^{\overline{\mathtt{d}}}$.
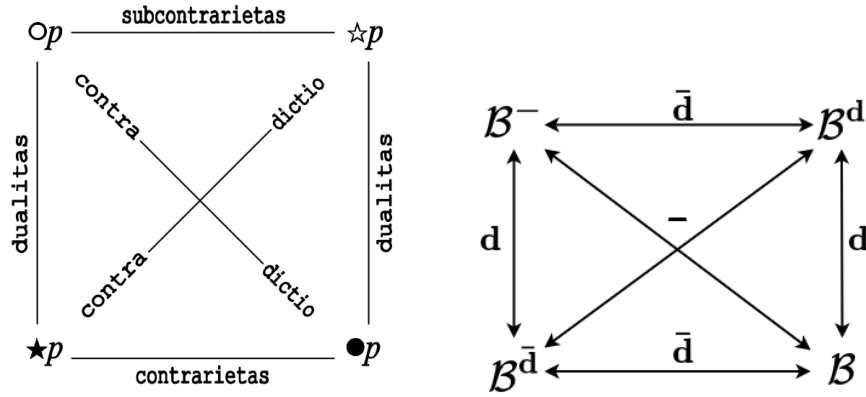
Fig. 13: On the left: Marcos' *square of perfections* [40] (edited and amended). On the right: rear face of the *topological cube of opposition* (as in Fig. 10).

Recall from the commutative diagram in Fig. 5 (the abstract *topological cube of opposition*) that $(\varphi^{\mathrm{d}})^{\mathrm{fp}} = (\varphi^{\overline{\mathrm{d}}})^{\overline{\mathrm{fp}}}$. This gives us another perspective to look at recovery operators. Let us now take an arbitrary negation(-like) operator η (playing the role of $\varphi^{\overline{\mathrm{d}}}$ above) and state again the generalized *principle of gentle explosion* and its dual:

$$\eta^{\overline{\mathrm{fp}}}(A) \wedge A \wedge \eta(A) = \bot \quad \text{dually} \quad \eta^{\overline{\mathrm{fp}}}(A) \leq A \vee \eta(A),$$

$$\eta^{\overline{\mathrm{fp}}}(A), A, \eta(A) \vdash \bot \quad \text{dually} \quad \vdash \eta^{\mathrm{fp}}(A), A, \eta(A).$$

In other words, the transformation $(\cdot)^{\overline{\mathrm{fp}}}$ behaves as a 'recovery meta-operator' that acts on an operator η (intended as a negation) and returns its corresponding recovery operator. In the case of the LFI and LFU operators discussed above, we can see that:

$$\circ = (\neg^{\mathcal{C}})^{\overline{\mathrm{fp}}} \quad \text{dually} \quad \star = (\neg^{\mathcal{I}})^{\overline{\mathrm{fp}}} \text{ (and so } \bigstar = (\neg^{\mathcal{I}})^{\mathrm{fp}}).$$

Last but not least, we remark that several properties of negation other than ECQ or TND can be recovered by applying fixed-point transformations to topological operators other than $\mathcal{I}$ and $\mathcal{C}$, such as $\mathcal{B}$ or $\mathcal{F}$, thus obtaining other, hitherto unnamed 'recovery-like' operators. For instance, the operator $\mathcal{B}^{\mathrm{fp}}(\cdot)$, while not able to recover TND or ECQ, can, under particular conditions, recover some contraposition, De Morgan and double-negation rules. Here again, we refer the reader to [26] for more details.

3.8 Quantifiers

We start by recalling from our discussion in §3.1 that quantifiers are introduced in HOL by means of the constants Π^{α} of type $(\alpha \Rightarrow o) \Rightarrow o$. They take a 'predicate' (i.e. a term of type $\alpha \Rightarrow o$) and return an appropriate 'truth-value' (type o), namely, true iff the predicate holds (is true) for *all* of the objects in α's domain.[47] Dually, the constants Σ^{α} can be

---

[47] A common definition using equality (and extensionality) is thus: $\Pi^{\alpha} \stackrel{\text{def}}{=} \lambda P. (P =^{\alpha \Rightarrow o} \lambda x. \text{true})$.

conveniently defined as: $\Sigma^\alpha \stackrel{\text{def}}{=} \lambda P. \neg(\Pi^\alpha (\lambda x. \neg(P\ x)))$. We shall recall that the only variable-binding mechanism available in HOL is that of λ-abstraction, so that the usual variable-binding quantifiers $\forall^\alpha$ and $\exists^\alpha$ are actually shorthand ('syntactic sugar').

$$\forall^\alpha x.\ P\ x \stackrel{\text{def}}{=} \Pi^\alpha (\lambda x.\ P) \qquad \exists^\alpha x.\ P\ x \stackrel{\text{def}}{=} \Sigma^\alpha (\lambda x.\ P)$$

In the spirit of shallow semantical embeddings, we introduce for the meta-logical terms Π^α and Σ^α their respective ω-type-lifted variants $\boldsymbol{\Pi}^\alpha$ and $\boldsymbol{\Sigma}^\alpha$ (in **boldface**) having as type: $(\alpha \Rightarrow (\omega)\sigma) \Rightarrow (\omega)\sigma$, i.e., they take a ω-type-lifted predicate (type $\alpha \Rightarrow (\omega)\sigma$) and return a ω-type-lifted 'truth-value' (type $(\omega)\sigma$).

$$\boldsymbol{\Pi}^\alpha\ \varphi \stackrel{\text{def}}{=} \lambda w.\ \forall^\alpha X.\ (\varphi\ X)\ w \qquad \boldsymbol{\Sigma}^\alpha\ \varphi \stackrel{\text{def}}{=} \lambda w.\ \exists^\alpha X.\ (\varphi\ X)\ w$$

We introduce for them convenient variable-binding notation $\boldsymbol{\forall}^\alpha$ and $\boldsymbol{\exists}^\alpha$.[48]

$$\boldsymbol{\forall} X.\ \varphi \stackrel{\text{def}}{=} \boldsymbol{\Pi} (\lambda X.\ \varphi) \qquad \boldsymbol{\exists} X.\ \varphi \stackrel{\text{def}}{=} \boldsymbol{\Sigma} (\lambda X.\ \varphi)$$

These quantifiers behave in the expected way, e.g., they are dual to each other.

$$\boldsymbol{\Pi}\ \varphi = -(\boldsymbol{\Sigma}\ \varphi^-) \qquad \text{i.e.}\ \ (\boldsymbol{\forall} X.\ \varphi\ X) = -(\boldsymbol{\exists} X.\ -(\varphi\ X))$$

Thus, using these quantifiers we can write first-order formulas like, e.g., the *drinker's principle* (and actually show it valid automatically).

$$\vdash \boldsymbol{\exists} x.\ \text{Drunk}\ x \rightarrow (\boldsymbol{\forall} y.\ \text{Drunk}\ y)$$

Our quantifiers can in fact be given an equivalent formulation in terms of infinitary supremum and infimum operations. Below we employ the notation $[\![\cdot]\!]$ to denote the range of a function, i.e., $[\![f]\!] \stackrel{\text{def}}{=} \lambda y.\ \exists x.\ (f\ x) = y$. Thus, we have that

$$\boldsymbol{\Pi}\ \varphi = \bigwedge [\![\varphi]\!] \qquad \boldsymbol{\Sigma}\ \varphi = \bigvee [\![\varphi]\!]$$

or using variable-binder notation

$$\boldsymbol{\forall} X.\ \varphi = \bigwedge [\![(\lambda X.\ \varphi)]\!] \qquad \boldsymbol{\exists} X.\ \varphi = \bigvee [\![(\lambda X.\ \varphi)]\!]\,.$$

Observe the similarity to the approach by Mostowski [42] of evaluating quantifiers as infinitary lattice operations. In fact, we have seen in §3.2 that our encoded Boolean algebras are lattice-complete, so that they naturally lend themselves to this sort of interpretation of quantifiers.[49]

In a similar spirit we can encode quantifiers whose domain of quantification is restricted to a fixed set of objects or individuals. We introduce a pair of quantifiers that

---

[48] Note the **boldface**. We omit the superscript α in the following. This kind of superscript-notation is also avoided in Isabelle/HOL, where *polymorphically-typed* constants are employed instead.

[49] We can also find this sort of interpretation for quantifiers in Boolean-valued models for set theory; see e.g. [5]. We think that our approach might also have an application in computer-supported (possibly non-classical) set-theoretical investigations.

take an additional domain set D (of α-parametric type $(\alpha)\sigma$) as a parameter. Hence the new quantifiers will have as a type: $(\alpha)\sigma \Rightarrow (\alpha \Rightarrow (\omega)\sigma) \Rightarrow (\omega)\sigma$.

$$\Pi[D]\,\varphi \stackrel{\text{def}}{=} \lambda w.\,\forall X.\,(D\ X) \longrightarrow (\varphi\ X)\ w \qquad \Sigma[D]\,\varphi \stackrel{\text{def}}{=} \lambda w.\,\exists X.\,(D\ X) \wedge (\varphi\ X)\ w$$

The restricted (or guarded) quantifiers introduced above are said to have a *constant domain*. In this respect they resemble their unconstrained counterparts, where the latter can be regarded as a special case of the former.

$$\Pi\,\varphi = \Pi[\top]\,\varphi \qquad\qquad \Sigma\,\varphi = \Sigma[\top]\,\varphi$$

In fact, restricted constant-domain quantifiers can analogously be characterized by means of infima and suprema.[50]

$$\Pi[D]\,\varphi = \bigwedge [\![\varphi\ D]\!] \qquad\qquad \Sigma[D]\,\varphi = \bigvee [\![\varphi\ D]\!]$$

Using the equivalences above, and observing that $[\![(f \circ g)\ S]\!] = [\![f\ [\![g\ S]\!]]\!]$, quantification for function composition can be stated in terms of restricted quantifiers.

$$\Pi(\varphi \circ \psi) = \Pi[[\![\psi]\!]]\,\varphi \qquad\qquad \Sigma(\varphi \circ \psi) = \Sigma[[\![\psi]\!]]\,\varphi$$

Other, more flexible kind of restricted (or guarded) quantifiers take a set-valued function $\delta(\cdot)$ (we may call it a 'domain function'), as an additional parameter of type $\alpha \Rightarrow (\omega)\sigma$. This function δ can be intuitively seen as mapping an (α-type-)object X to the proposition "X exists".[51] Hence varying-domain quantifiers have type $(\alpha \Rightarrow (\omega)\sigma) \Rightarrow (\alpha \Rightarrow (\omega)\sigma) \Rightarrow (\omega)\sigma$.

$$\Pi\{\delta\}\,\varphi \stackrel{\text{def}}{=} \lambda w.\,\forall X.\,(\delta\ X)\ w \longrightarrow (\varphi\ X)\ w \qquad \Sigma\{\delta\}\,\varphi \stackrel{\text{def}}{=} \lambda w.\,\exists X.\,(\delta\ X)\ w \wedge (\varphi\ X)\ w$$

Varying-domain quantifiers are dual to each other in the expected way, i.e.,

$$\Pi\{\delta\}\,\varphi = -(\Sigma\{\delta\}\,\varphi^-).$$

It is easily seen that varying-domain quantification generalizes its constant-domain counterpart (below we employ the notation $f\!\uparrow \stackrel{\text{def}}{=} \lambda x.\,\lambda y.\,f\ x$).[52]

$$\Pi[D]\,\varphi = \Pi\{D\!\uparrow\}\,\varphi \qquad\qquad \Sigma[D]\,\varphi = \Sigma\{D\!\uparrow\}\,\varphi$$

Recalling that set-valued functions (of type $\alpha \Rightarrow (\omega)\sigma$) form a Boolean algebra whose top element is denoted by $\ddot{\top}$, we can easily present unrestricted quantifiers as a special case of varying-domain quantifiers.

$$\Pi\,\varphi = \Pi\{\ddot{\top}\}\,\varphi \qquad\qquad \Sigma\,\varphi = \Sigma\{\ddot{\top}\}\,\varphi$$

---

[50] Recall that we employ the notation $[\![\cdot\ \cdot]\!]$ to denote the image of a set under a function, i.e., $[\![f\ S]\!] \stackrel{\text{def}}{=} \lambda y.\,\exists x.\,(S\ x) \wedge (f\ x) = y$.

[51] In line with what some refer to as a (meta-logical) 'existence' predicate, which is used to restrict the domains of quantification in presentations of varying-domains semantics for quantified modal logics (cf. [25,24] for an exposition).

[52] In general $f\!\uparrow$ can be seen as a sort of 'inverse projection' that lifts a unary function f to a binary function that is 'projected' wrt. its first argument. This is useful, e.g., for converting a set D into a 'rigid' set-valued function $\delta = D\!\uparrow$.

In fact, we can employ other Boolean connectives on set-valued functions to encode varying-domain quantifiers as unrestricted ones.

$$\mathbf{\Pi}\{\delta\}\,\varphi = \mathbf{\Pi}(\delta \mathbin{\ddot{\rightarrow}} \varphi) \qquad\qquad \mathbf{\Sigma}\{\delta\}\,\varphi = \mathbf{\Sigma}(\delta \mathbin{\ddot{\curlywedge}} \varphi)$$

In this way, different sorts of restricted quantification can be equivalently encoded using unrestricted variable-binding quantifiers, and employing the connectives $\ddot{\rightarrow}$ and $\ddot{\curlywedge}$ to adequately relativize predicates.

$$\mathbf{\Sigma}\{\delta\}\,\varphi = \forall X.\,(\delta \mathbin{\ddot{\curlywedge}} \varphi)\,X \qquad\qquad \mathbf{\Pi}[D]\,\varphi = \forall X.\,(D{\upharpoonright} \mathbin{\ddot{\rightarrow}} \varphi)\,X$$

Propositional and Higher-order Quantifiers. The reader may have noticed that in our previous discussion we have not specified at which 'order' our quantifiers operate, i.e., whether they are intended as first-order, propositional or higher-order quantifiers. In fact, our (α-parametric) quantifiers behave as first-order quantifiers when instantiating the type parameter α with some type for 'individuals', say ι. They behave as propositional quantifiers when α is instantiated with type $(\omega)\sigma$. Different sorts of higher-order quantification can be obtaining by instantiating α with the appropriate type, e.g., with $\iota\Rightarrow(\omega)\sigma$ for quantifying over (intensional) predicates.[53] In the case of, e.g., propositional quantification, the formula below is well-formed (and easily shown valid automatically):

$$\vdash \forall A.\,(\exists B.\,A \leftrightarrow -B)\,.$$

An interesting application in non-classical logics arises when we employ quantifiers restricted to domains of fixed points of operators. For the sake of illustration, let us assume below that $\mathcal{I}$ is an operator that satisfies the (duals of) Kuratowski conditions, i.e., $\mathcal{I}$ is a topological interior operator. If we employ quantifiers restricted to the set of $\mathcal{I}$'s fixed points (aka. open sets) we have that in fact the previous formula does not hold:

$$\nvdash \forall^{\mathcal{I}} A.\,(\exists^{\mathcal{I}} B.\,A \leftrightarrow -B)\,,$$

where $\forall^{\mathcal{I}} X.\,\varphi = \mathbf{\Pi}[\mathtt{fp}\,\mathcal{I}]\,(\lambda X.\,\varphi)$ and $\exists^{\mathcal{I}} X.\,\varphi = \mathbf{\Sigma}[\mathtt{fp}\,\mathcal{I}]\,(\lambda X.\,\varphi)$.

In fact, one can reason in intuitionistic logic using our approach by replacing all (implicit[54]) propositional quantifiers in formulas by the ones defined above and employing the paracomplete negation $\neg^{\mathcal{I}} \stackrel{\text{def}}{=} \mathcal{I}^{\bar{\mathsf{d}}}$ instead of the classical one (implication needs to be suitably translated[55]). This derives from the fact that the algebra of open sets of a topological space is a Heyting algebra. The same approach applies *mutatis mutandis* to other non-classical logics with (not necessarily Boolean) lattice-based semantics. They shall be investigated in future follow-up work.

---

[53] This makes best sense in the context of an *intensional* higher-order logic; see [24] for a thorough discussion and [28] for its encoding in Isabelle/HOL.

[54] For instance, we need to take the universal closure (using $\forall^{\mathcal{I}}$) wrt. all schema/free variables.

[55] Since classical implication does not preserve open sets it needs to be suitably encoded, e.g., as done in Gödel's translation of intuitionistic logic into **S4**.

The Barcan Formula and its Converse No discussion of quantification in non-classical contexts would be complete without revisiting the Barcan formula (and its converse). We start by noting that the converse Barcan formula follows readily from monotonicity for any operator φ (wrt. any arbitrary predicate ψ of type $\alpha{\Rightarrow}(\omega)\sigma$).

$$\texttt{MONO}\ \varphi \longrightarrow \varphi\,(\forall x.\ \psi\ x) \le \forall x.\ \varphi\,(\psi\ x) \qquad \text{(CBF-1)}$$

$$\texttt{MONO}\ \varphi \longrightarrow \exists x.\ \varphi\,(\psi\ x) \le \varphi\,(\exists x.\ \psi\ x) \qquad \text{(CBF-2)}$$

So it is in fact the Barcan formula which requires stronger assumptions (of an 'infinitary character') in order to hold.

$$\texttt{iMULT}^{\text{b}}\ \varphi \longrightarrow \forall x.\ \varphi\,(\psi\ x) \le \varphi\,(\forall x.\ \psi\ x) \qquad \text{(BF-1)}$$

$$\texttt{iADDI}^{\text{a}}\ \varphi \longrightarrow \varphi\,(\exists x.\ \psi\ x) \le \exists x.\ \varphi\,(\psi\ x) \qquad \text{(BF-2)}$$

Regarding restricted quantification, we observe a difference in the behavior of varying-domain quantifiers in contrast to their constant-domain counterparts. The former generally do not validate the Barcan formula nor its converse, while the latter behave in the same way as their unrestricted counterparts. For an illustration, let us take the universally-quantified variants for the Barcan formula and its converse, BF-1 resp. CBF-1. First, we repeat the results above for the unrestricted case (recalling that $\texttt{MONO}\ \varphi$ iff $\texttt{iMULT}^{\text{a}}\ \varphi$), but this time without employing variable-binding notation for quantifiers.

$$\texttt{iMULT}^{\text{b}}\ \varphi \longrightarrow \boldsymbol{\Pi}\,(\varphi \circ \psi) \le \varphi\,(\boldsymbol{\Pi}\,\psi) \qquad \text{(BF-1')}$$

$$\texttt{iMULT}^{\text{a}}\ \varphi \longrightarrow \varphi\,(\boldsymbol{\Pi}\,\psi) \le \boldsymbol{\Pi}\,(\varphi \circ \psi) \qquad \text{(CBF-1')}$$

The results above hold for quantifiers restricted to constant domains too. The HOL formulas below are thus valid (for an arbitrary domain set D of type $(\alpha)\sigma$).

$$\texttt{iMULT}^{\text{b}}\ \varphi \longrightarrow \boldsymbol{\Pi}[D]\,(\varphi \circ \psi) \le \varphi\,(\boldsymbol{\Pi}[D]\,\psi) \qquad \text{(BF-1-cons)}$$

$$\texttt{iMULT}^{\text{a}}\ \varphi \longrightarrow \varphi\,(\boldsymbol{\Pi}[D]\,\psi) \le \boldsymbol{\Pi}[D]\,(\varphi \circ \psi) \qquad \text{(CBF-1-cons)}$$

However, in the case of varying domains the results above do not hold. The HOL formulas below are countersatisfiable (for an arbitrary domain function δ of type $\alpha{\Rightarrow}(\omega)\sigma$). We have obtained countermodels using Isabelle's model generation tool *Nitpick* [13] even by assuming all (infinitary) interior conditions on φ.

$$\texttt{iMULT}\ \varphi \wedge \texttt{CNTR}\ \varphi \wedge \texttt{IDEM}\ \varphi \not\longrightarrow \boldsymbol{\Pi}\{\delta\}\,(\varphi \circ \psi) \le \varphi(\boldsymbol{\Pi}\{\delta\}\,\psi) \qquad \text{(BF-1-var)}$$

$$\texttt{iMULT}\ \varphi \wedge \texttt{CNTR}\ \varphi \wedge \texttt{IDEM}\ \varphi \not\longrightarrow \varphi(\boldsymbol{\Pi}\{\delta\}\,\psi) \le \boldsymbol{\Pi}\{\delta\}\,(\varphi \circ \psi) \qquad \text{(CBF-1-var)}$$

The reason why the Barcan formula and its converse hold (under the appropriate conditions) for constant-domain quantifiers (including unrestricted ones) becomes evident if we recall, from the previous subsection, the characterization of constant-domain quantification in terms of infinitary infima and suprema.

$$\boldsymbol{\Pi}[D]\,(\varphi \circ \psi) = \bigwedge [\![(\varphi \circ \psi)\ D]\!] \quad \text{and} \quad \boldsymbol{\Pi}[D]\,\psi = \bigwedge [\![\psi\ D]\!]$$

We can now state `BF-1-cons` and `CBF-1-cons` somehow more explicitly.

$$\texttt{iMULT}^{\text{b}}\,\varphi \longrightarrow \bigwedge [\![(\varphi \circ \psi)\,D]\!] \leq \varphi\,(\bigwedge [\![\psi\,D]\!]) \qquad (\texttt{BF-1-cons'})$$

$$\texttt{iMULT}^{\text{a}}\,\varphi \longrightarrow \varphi\,(\bigwedge [\![\psi\,D]\!]) \leq \bigwedge [\![(\varphi \circ \psi)\,D]\!] \qquad (\texttt{CBF-1-cons'})$$

Finally, unfolding the definitions for $\texttt{iMULT}^{\text{a}}$ and $\texttt{iMULT}^{\text{b}}$ above, and observing that $[\![(\varphi \circ \psi)\,D]\!] = [\![\varphi\,[\![\psi\,D]\!]]\!]$, we can state `BF-1-cons` and `CBF-1-cons` now in a very explicit fashion.

$$(\forall S.\, \bigwedge [\![\varphi\,S]\!] \leq \varphi(\bigwedge S)) \longrightarrow \bigwedge [\![\varphi\,[\![\psi\,D]\!]]\!] \leq \varphi\,(\bigwedge [\![\psi\,D]\!]) \qquad (\texttt{BF-1-cons''})$$

$$(\forall S.\, \varphi(\bigwedge S) \leq \bigwedge [\![\varphi\,S]\!]) \longrightarrow \varphi\,(\bigwedge [\![\psi\,D]\!]) \leq \bigwedge [\![\varphi\,[\![\psi\,D]\!]]\!] \qquad (\texttt{CBF-1-cons''})$$

Thus, we can observe how, in the case of constant-domain quantification, the Barcan formula and its converse become straightforwardly entailed by $\texttt{iMULT}^{\text{b}}$ and $\texttt{iMULT}^{\text{a}}$ respectively (or, dually, by $\texttt{iADDI}^{\text{a}}$ and $\texttt{iADDI}^{\text{b}}$ respectively). However, the same cannot be said in the context of varying-domain quantification. Loosely speaking, varying-domain quantifiers cannot be 'squeezed' into a characterization in terms of infinitary infima and suprema, as in the case of their constant-domain counterparts. They feature an additional 'degree of freedom', with their domain of quantification being restricted by a set-valued function rather than just a set.

4 Conclusion and Prospects

We employed a shallow semantical embedding approach [6,10] to encode the key concepts of a broad theory of topological Boolean algebras (cf. early works by Kuratowski, Zarycki, McKinsey & Tarski, among others). We utilized this to provide a natural semantics for *quantified* paraconsistent and paracomplete logics featuring *recovery operators*, particularly certain families of Logics of Formal Inconsistency (LFIs) [18,15,17] and Logics of Formal Undeterminedness (LFUs) [40,17]. We demonstrated how our method facilitates a uniform characterization of propositional, first-order, and higher-order quantification (restricted also to constant and varying domains). The algebraic semantics presented in this paper has, in fact, grown out of an effort to generalize the neighborhood structures introduced in [16, §5].

A main contribution of our work thus consists in showing how paraconsistent and paracomplete negations, along with recovery operators, can be explicated in terms of (generalized) topological operators, among them modalities (by recalling the notions of Alexandrov topologies and specialization preorders, cf. §3.4). The idea of defining non-classical negations by composing a classical negation with a modality is of course not new and has been suggested in numerous occasions by researchers working at the crossroads of modal and non-classical logics. The related papers are too many for us to provide an adequate (and fair) survey. We restrict ourselves to giving a special mention to the works [40,22] (and references therein) for modal-logic-oriented approaches, and [4,21] (and references therein) for topologically-oriented approaches relevant to our work. In fact, the latter paper by Coniglio & Prieto-Sanabria [21] is very close in spirit to

ours, although the scope of their results differs. Future computer-assisted investigations will strive to bridge their findings with ours.

An additional contribution involves providing a semantically motivated characterization of quantifiers for the non-classical logics under study. We have begun examining their basic interactions with other connectives, but there is much more to explore. Leveraging computational tools, we aim to deepen our understanding of the properties of suitably interrelated (combinations of) primitive operators, as exemplified in Rauszer's *bi-topological Boolean algebras* [45]. We hypothesize that numerous negation(-like) operators examined in the literature can be defined this way (potentially paired with their respective recovery operators). Computer-assisted experiments will be instrumental in shedding further light on their interactions with various types of quantifiers.

So far, the concept of recovery operators has been discussed within the context of paraconsistent and paracomplete logics. Therefore, we have initiated our investigations with the paradigmatic case of negations (and associated modalities). Future work will explore other (binary) connectives. Further generalization of the present theory to provide an unifying semantics for various other systems of non-classical and substructural logics (possibly including relevance and linear logic) is the most likely medium-term outcome of this work. This involves suitably extending the *algebra of operators* presented in §3.3. In the long term, our objective is to provide symbolic ethico-legal reasoning capabilities for AI systems. The present formalization [26] complements other works based on the *LogiKEy* framework and methodology [9] (`logikey.org`). Our work could also be of interest to the working logician or mathematician, who might appreciate tedious pen-and-paper calculations being replaced by invocations of model generators (e.g., *Nitpick* [13]) or automated theorem provers (via *Sledgehammer* [12]). Notably, besides their traditional use in formalized mathematics, proof assistants have also been successfully employed in the formal verification of hardware and software systems. One of our goals is thus to apply techniques from formal verification, particularly those using HOL-based systems, to the development of AI systems with logico-pluralistic reasoning capabilities.

Our work has been primarily oriented towards semantics. However, proof-theoretical investigations with the aim of developing sound and complete calculi represent significant potential follow-up work. We are currently exploring techniques to utilize mathematical proof assistants, particularly Isabelle/HOL, for formally verifying proof calculi for non-classical logical systems. Indeed, fully automatic verification of calculi soundness is straightforward using the shallow semantical embedding approach; it simply requires that we check whether the corresponding rules, encoded as meta-logical implications, follow as theorems from the axiomatic semantic (e.g., frame) conditions plus the definitions of object-logical connectives. Completeness proofs require, however, an explicit encoding of the syntax and rules of the object calculi,[56] together with elaborated 'quotient' constructions, which demand relatively complex proofs by induction whose full automation remains an open problem [36]. Assuming, as we do, that our object logics are defined semantically via their shallow embeddings (wrt. HOL as a meta-language), we only need to allow for digressions into meta-logical reasoning by suitable provers (complete wrt. HOL [8,1]) during proof search to ensure completeness.

---

[56] This corresponds to the notion of a *deep embedding*. We refer the reader to [30] for a discussion of the differences (and similarities) between *deep* and *shallow* embeddings.

References

1. Andréka, H., van Benthem, J., Bezhanishvili, N., Németi, I.: Changing a semantics: Opportunism or courage? In: Manzano, M., Sain, I., Alonso, E. (eds.) The Life and Work of Leon Henkin: Essays on His Contributions, pp. 307–337. Springer International Publishing, Cham (2014)

2. Andrews, P.B.: General models and extensionality. The Journal of Symbolic Logic **37**(2), 395–397 (1972)

3. Andrews, P.B.: An introduction to mathematical logic and type theory: to truth through proof, Applied Logic Series, vol. 27. Springer Dordrecht, 2 edn. (2002)

4. Başkent, C.: Some topological properties of paraconsistent models. Synthese **190**(18), 4023–4040 (2013)

5. Bell, J.L.: Set Theory: Boolean-Valued Models and Independence Proofs. Oxford Logic Guides, Oxford Scuence Publications (2005)

6. Benzmüller, C.: Universal (meta-)logical reasoning: Recent successes. Science of Computer Programming **172**, 48–62 (2019)

7. Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. Journal of Symbolic Logic **69**(4), 1027–1088 (2004)

8. Benzmüller, C., Miller, D.: Automation of higher-order logic. In: Gabbay, D.M., Siekmann, J.H., Woods, J. (eds.) Handbook of the History of Logic, Volume 9 — Computational Logic, pp. 215–254. North Holland, Elsevier (2014)

9. Benzmüller, C., Parent, X., van der Torre, L.: Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. Artificial Intelligence **287**, 103348 (2020)

10. Benzmüller, C., Paulson, L.C.: Quantified multimodal logics in simple type theory. Logica Universalis (Special Issue on Multimodal Logics) **7**(1), 7–20 (2013)

11. Benzmüller, C., Andrews, P.: Church's Type Theory. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Summer 2019 edn. (2019), `https://plato.stanford.edu/archives/sum2019/entries/type-theory-church/`

12. Blanchette, J.C., Kaliszyk, C., Paulson, L.C., Urban, J.: Hammering towards QED. Journal of Formalized Reasoning **9**(1), 101–148 (2016)

13. Blanchette, J.C., Nipkow, T.: Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In: Kaufmann, M., Paulson, L.C. (eds.) ITP 2010. LNCS, vol. 6172, pp. 131–146. Springer (2010)

14. Bou, F., Esteva, F., Font, J.M., Gil, À.J., Godo, L., Torrens, A., Verdú, V.: Logics preserving degrees of truth from varieties of residuated lattices. Journal of Logic and Computation **19**(6), 1031–1069 (2009)

15. Carnielli, W.A., Coniglio, M.E.: Paraconsistent Logic: Consistency, Contradiction and Negation, Logic, Epistemology, and the Unity of Science, vol. 40. Springer, Cham (2016)

16. Carnielli, W.A., Coniglio, M.E., Fuenmayor, D.: Logics of formal inconsistency enriched with replacement: An algebraic and modal account. The Review of Symbolic Logic **15**(3), 771–806 (2022)

17. Carnielli, W.A., Coniglio, M.E., Rodrigues, A.: Recovery operators, paraconsistency and duality. Logic Journal of the IGPL **28**(5), 624–656 (2020)

18. Carnielli, W.A., Marcos, J.: A taxonomy of C-systems. In: Carnielli, W.A., Coniglio, M.E., D'Ottaviano, I.M.L. (eds.) Paraconsistency: The Logical Way to the Inconsistent. Proceedings of the 2nd World Congress on Paraconsistency (WCP 2000), Lecture Notes in Pure and Applied Mathematics, vol. 228, pp. 1–94. Marcel Dekker, New York (2002)

19. Carnielli, W.A., Rodrigues, A.: An epistemic approach to paraconsistency: a logic of evidence and truth. Synthese **196**(9), 3789–3813 (2017)
20. Church, A.: A formulation of the simple theory of types. J. Symb. Log. **5**(2), 56–68 (1940)
21. Coniglio, M.E., Prieto-Sanabria, L.: Modal logic S4 as a paraconsistent logic with a topological semantics. In: P. Gouveia, C.C., Donisio, F. (eds.) Logic and Computation: Essays in Honour of Amilcar Sernadas, Tributes, vol. 33, pp. 171–196. College Publications, London (2017)
22. Dodó, A., Marcos, J.: Negative modalities, consistency and determinedness. Electronic Notes in Theoretical Computer Science **300**, 21–45 (2014)
23. Esakia, L.: Intuitionistic logic and modality via topology. Annals of Pure and Applied Logic **127**(1-3), 155–170 (2004)
24. Fitting, M.: Types, Tableaus, and Gödel's God, Trends in logic, vol. 12. Springer Science & Business Media (2002)
25. Fitting, M., Mendelsohn, R.L.: First-order modal logic, Synthese Library, vol. 277. Springer Science & Business Media (1998)
26. Fuenmayor, D.: Topological semantics for paraconsistent and paracomplete logics. Archive of Formal Proofs (2020), (`https://isa-afp.org/entries/Topological_Semantics.html`. Check instead `https://github.com/davfuenmayor/topological-semantics` for an up-to-date development version of this entry)
27. Fuenmayor, D., Benzmüller, C.: Normative reasoning with expressive logic combinations. In: De Giacomo, G., Catala, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) ECAI 2020 – 24th European Conference on Artificial Intelligence, June 8-12, Santiago de Compostela, Spain. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2903–2904. IOS Press (2020)
28. Fuenmayor, D., Benzmüller, C.: Types, Tableaus and Gödel's God in Isabelle/HOL. Archive of Formal Proofs (May 2017), `https://isa-afp.org/entries/Types_Tableaus_and_Goedels_God.html`, Formal proof development
29. Fuenmayor, D., Serrano Suárez, F.F.: Formalising basic topology for computational logic in simple type theory. In: Buzzard, K., Kutsia, T. (eds.) Intelligent Computer Mathematics. pp. 56–74. Springer International Publishing, Cham (2022)
30. Gibbons, J., Wu, N.: Folding domain-specific languages: deep and shallow embeddings (functional pearl). In: Jeuring, J., Chakravarty, M.M.T. (eds.) Proceedings of the 19th ACM SIGPLAN international conference on Functional programming, Gothenburg, Sweden, September 1-3, 2014. pp. 339–347. ACM (2014)
31. Gordon, M.J.: HOL: A proof generating system for higher-order logic. In: VLSI specification, verification and synthesis, pp. 73–128. Springer (1988)
32. Gordon, M.: HOL - a machine oriented formulation of higher order logic. Technical Report 68, University of Cambridge - Computer Laboratory (July 1985)
33. Harrison, J.: HOL light: An overview. In: International Conference on Theorem Proving in Higher Order Logics. pp. 60–66. Springer (2009)
34. Hausdorff, F.: Grundzüge der Mengenlehre, vol. 7. von Veit (1914)
35. Henkin, L.: Completeness in the theory of types. The Journal of Symbolic Logic **15**(2), 81–91 (1950)
36. Johansson, M.: Lemma discovery for induction: A survey. In: Intelligent Computer Mathematics: 12th International Conference, CICM 2019, Prague, Czech Republic, July 8–12, 2019, Proceedings 12. pp. 125–139. Springer (2019)
37. Jónsson, B., Tarski, A.: Boolean algebras with operators. Part I. American journal of mathematics **73**(4), 891–939 (1951)
38. Kuratowski, K.: Sur l'opération Ā de l'analysis situs. Fundamenta Mathematicae **3**(1), 182–199 (1922)

39. Kuratowski, K.: Topologie I, vol. 20. Monografie Matematyczne (1948)
40. Marcos, J.: Nearly every normal modal logic is paranormal. Logique et Analyse **48**(189/192), 279–300 (2005)
41. McKinsey, J.C., Tarski, A.: The algebra of topology. Annals of mathematics **45**, 141–191 (1944)
42. Mostowski, A.: Proofs of non-deducibility in intuitionistic functional calculus. The Journal of Symbolic Logic **13**(4), 204–207 (1948)
43. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, LNCS, vol. 2283. Springer (2002)
44. Rasiowa, H., Sikorski, R.: The Mathematics of Metamathematics. Panstwowe Wydawnictwo Naukowe (1963)
45. Rauszer, C.: Semi-boolean algebras and their applications to intuitionistic logic with dual operations. Fundamenta Mathematicae **83** (1974)
46. Schönfinkel, M.: Über die Bausteine der mathematischen Logik. Mathematische Annalen **92**, 305–316 (1924)
47. Tarski, A.: The concept of truth in the languages of the deductive sciences. Prace Towarzystwa Naukowego Warszawskiego, Wydzial III Nauk Matematyczno-Fizycznych **34**(13-172), 198 (1933)
48. Zarycki, M.: Quelques notions fondamentales de l'analysis situs au point de vue de l'algèbre de la logique. Fundamenta Mathematicae **9**(1), 3–15 (1927)
49. Zarycki, M.: Allgemeine Eigenschaften der cantorschen Kohärenzen. Transactions of the American Mathematical Society **30**(3), 498–506 (1928)
50. Zarycki, M.: Some properties of the derived set operation in abstract spaces. Nauk. Zap. Ser. Fiz.-Mat. **5**, 22–33 (1947)