Pere Pardo

A Modal View on Resource-Bounded Propositional Logics

Abstract. Classical propositional logic plays a prominent role in industrial applications, and yet the complexity of this logic is presumed to be non-feasible. Tractable systems such as depth-bounded boolean logics approximate classical logic and can be seen as a model for resource-bounded agents whose reasoning style is nonetheless classical. In this paper we first study a hierarchy of tractable logics that is not defined by depth. Then we extend it into a modal logic where modalities make explicit the assumptions discharged in propositional proofs, thereby expressing blueprints for proofs. A natural deduction system is provided that permits to reason about and manage such proof blueprints.

Keywords: Resource-bounded logic, Classical propositional logic, Modal logic, Bivalence rule, Information semantics, Distributed theorem proving.

Introduction

Classical propositional logic plays a prominent role in many industrial applications [27,34]. Its satisfiability and validity problems, though, are NP-complete resp. coNP-complete [8] and so under the widely accepted conjecture that $\mathsf{P} \neq \mathsf{NP}$, classical reasoning would lie beyond what is considered computationally feasible. Resource-bounded logics for propositional languages [10,11,13,15,18,19,39,40] address these concerns and offer more realistic models of reasoning agents.[1] These logics are obtained by setting bounds on certain inference rules of the proof system $\vdash$, thereby trading deductive power (completeness of $\vdash$) for a lower complexity. This is a useful abstraction for modelling physical bounds on the memory and time available to agents —and even perhaps accurate for human reasoners. More importantly, propositional resource-bounded logics such as DBBL are tractable [10]. In the present paper, we focus on the informational approach to classical

[1] Classical and epistemic logics suffer from *logical omniscience* problems as models of reasoning agents [46]. Epistemic logic [44] is thus said to model what an agent *can* know, or does *implicitly* know, by purely deductive methods.

Presented by **Jacek Malinowski**; *Received* November 30, 2020

Studia Logica
https://doi.org/10.1007/s11225-022-09984-3

Published online: 07 April 2022

logic [10,11] and study a modal extension for reasoning about blueprints for propositional proofs. Our motivation lies in distributed reasoning applications, such as query-answering and consistency checks in a shared or distributed data base. On this topic, a fundamental question is what exactly is to be communicated by reasoning agents aimed at such tasks.

A list of desiderata for such a *deductive exchange* format are: (i) reusability, (ii) fault-tolerance, (iii) optimality preservation, and (iv) succinctness. *Reusability* is a bare minimum condition: being able to reach further sound conclusions, which amounts to soundness preservation. *Tolerance to faults* generated at proof search, storage or messaging is here understood as the existence of quick checks on correctness. *Optimality preservation* prevents the introduction of new unnecessary steps or assumptions after any deductive exchange.[2] Among message forms meeting conditions (i)–(iii), one should opt for (iv) a *succint* form requiring less memory use and messaging time.

Given a proof Π, say witnessing that $\Delta \vdash B$, a deductive exchange format can be anything between: the full proof Π or just the proof conclusion B. These two extreme cases do not fulfil the above desiderata: messaging full proofs [3,36] satisfies all conditions except for succintness, as many proof steps can be omitted without a significant loss w.r.t. (ii)–(iii); conclusions, in turn, are maximally succinct but fail on (ii) as correctness tests amount to new proofs from scratch; messaging conclusions might also fail on (iii) as merging optimal proofs (using cut) might lead to redundant subproofs or assumptions. To better fulfil these desiderata, we introduce a notion of *proof blueprint*.

Natural deduction for classical logic is presented in [10,11] as a system $\vdash$ with a unique rule RB for discharging assumptions (in pairs) $\{A, \neg A\}$ (Figure 1, left). The analysis of proof complexity in [10, p. 87] identifies the search space of RB-assumptions as the key element for the complexity of deciding whether $\Gamma \vdash B$. For distributed proofs, these insights suggest a deductive exchange format consisting of (i) the premises $\Delta \subseteq \Gamma$ used in some proof of $\Delta \vdash B$ and (ii) the formula pairs $\{A, \neg A\}$ discharged by RB as a sort of *blueprint* for this proof —other proof steps being completely mechanical.[3] Actually, we build this notion upon a hierarchy of tractable logics, called

[2] Hence, if a node sends an optimal proof (containing no redundancies or unnecessary assumptions), the receiving node must be able to reconstruct essentially the same proof. While it is trivial to check that all assumptions in a proof are actually used, checking that these are minimal for the given conclusion is not.

[3] This mechanical character is apparent from depth-bounded boolean logics enjoying the *weak subformula property*, so proofs of $\Gamma \vdash A$ exist with only weak subformulas of $\Gamma \cup \{A\}$. See [13] for proof transformations that delete detours and other redundancies.

$$\dfrac{\begin{array}{cc}\{A\} & \{\neg A\}\\ \vdots & \vdots \\ B & B\end{array}}{B}\;RB \qquad\qquad \dfrac{\begin{array}{c}\{A\}\\ \vdots\\ B\end{array}}{A\to B}\;I{\to}$$

Figure 1. (Left) An application of the Bivalence Rule: if B can be inferred under the assumption that A and also under the assumption that $\neg A$, then infer B. Both assumptions are discharged (denoted with braces) by this application of RB. (Right) A rule for the introduction of implication used commonly in natural deduction for classical logic [37]

UBBL, that measures proof complexity by the number of RB-formulas, rather than their depth as in [11]. This measure, moreover, enables a modal logic for reasoning *about* proof blueprints. In this logic, a formula $[\pm A_1]\ldots[\pm A_k]B$ represents that a classical proof of B exists whose RB-instances involve at most the pairs $\{A_1, \neg A_1\}, \ldots \{A_k, \neg A_k\}$. This modal logic can equip agents with tools for merging and optimizing blueprints.

Related work Different hierarchies of classically sound but incomplete systems, called resource-bounded logics, have been shown to approximate classical logic. Under a 2-valued semantics, Schaerf and Cadoli [39] present a chain of logics $\models_S$ that validate *modus ponens* for increasing sets S of atoms; its valuations can be paraconsistent for $p \notin S$, i.e. $V(p) = 1 = V(\neg p)$.[4] In the same vein, Finger and Wassermann [19] describe a general system that validates *each rule* ρ within a set of applicable formulas S_ρ only; these sets S_ρ (and so the logic) can expand during proof search, thus imposing proof heuristics based on rule precedence. In an algebraic setting, Dalal [15] builds upon boolean constraint propagation to bound the size of the *cut formula*. This is generalized in the lattice approach of Finger and Gabbay [18] built on an infinite layer of truth-values between 0 and 1; the *cut rule* is not eliminable and sequent calculi bound its use by the number of atoms or formulas. Using a 3-valued non-deterministic semantics, D'Agostino et al. [10,11,13] define the DBBL hierarchy by bounds on the *nesting of* RB, its only branching rule. The same bounded logics are found in Stålmarck's method [40], which further extends them with rules for truth-value equivalences $A \equiv B$, among other novel features (see [11,40] for a comparison). For resource-bounded logics in first-order languages, an overview can be found in [23].

[4] A dual approximation to classical logic is in fact presented in [39], by simultaneously defining a hierarchy (from above) of unsound and complete logics $\models'_S$ for disproving validity claims. Here, valuations can be paracomplete for $p \notin S$, i.e. $V'(p) = 0 = V'(\neg p)$.

Epistemic logics have also been designed to avoid the logical omniscience problem [46]. These proposals distinguish explicit from implicit knowledge either by weakening axiom K [45] or by modelling reasoning steps dynamically or as the result of the passing of time or a number of derivation steps [1,2,5,30,42]. Hawke et al. [28] discusses the plausibility of modal versions of classical inferences for resource-bounded agents, in partial disagreement with D'Agostino [10] or the present approach. A related approach is that of logics of awareness, see [25,43] for recent work along this line. The above described [39] also considers modalities $\Box_S$ for knowledge in the paraconsistent logics $\models_S$. In comparison, our dynamic modalities validate the axioms for knowledge from Schaerf and Cadoli [39] plus some interaction axioms; their modular character also suits better applications in distributed reasoning. Along this line, epistemic justification logics [3,36] describe modalities (proof terms) that encode full proofs of the formulas under their scope, similarly to proof blueprints. These modal logics mostly build upon classical propositional logic, unlike the present work or Klassen [31], which describes a multi-agent belief logic with separate update modalities $[A], [\neg A]$ and Kleene's semantics (see footnote 6).

Finally, modal logics of distributed knowledge address what agents do know as a group [26,44]. Recent extensions of DBBL with multi-agent [7] and dynamic epistemic [33] modalities also address logical omniscience. Distributed reasoning in formal argumentation has been considered with DBBL in place of classical logic [14]. For first-order languages, a natural deduction system for distributed reasoning is considered in [22].

Structure of the paper Section 1 starts by recalling depth-bounded boolean logics DBBL. Section 2 studies a measure of proof complexity based on the number of RB-formulas used. Section 3 presents the lattice of valuations. This and Section 4 on models for 0-depth logic pave the way for degree-bounded logics UBBL. Section 5 studies applications of RB as semantic updates. Section 6 introduces the UBBL hierarchy and Section 7 presents a modal logic with update modalities. Section 8 describes a complete deductive system for this logic. Section 9 addresses the complexity of its validity problem and discusses applications in distributed reasoning. The paper concludes with directions for future research. Appendix A recalls fundamental facts from lattice theory and Appendix B contains proofs of some results.

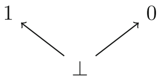

Figure 2. The information order $\leq$ on truth-values $\bot \leq 1$ and $\bot \leq 0$ according to the information they carry: from less to more information

$\wedge$	1	0	$\bot$
1	1	0	$\bot$
0	0	0	0
$\bot$	$\bot$	0	$0, \bot$

$\vee$	1	0	$\bot$
1	1	1	1
0	1	0	$\bot$
$\bot$	1	$\bot$	$1, \bot$

$\rightarrow$	1	0	$\bot$
1	1	0	$\bot$
0	1	1	1
$\bot$	1	$\bot$	$1, \bot$

$\neg$	
1	0
0	1
$\bot$	$\bot$

Figure 3. Truth-tables f_* for $* \in \{\wedge, \vee, \neg, \rightarrow\}$. See [11] for other connectives and for an intuitive explanation of the truth-tables in terms of information modules

1. Preliminaries: DBBL Depth-Bounded Boolean Logics

Let us start with a reminder of 0-depth logic and the inductive definition of k+1-depth logics that together define the DBBL logical hierarchy [10].

Definition 1.1. (*Propositional language* $\mathcal{L}$) Given a set of atoms $\mathsf{Var} = \{p_1, \ldots, p_i, \ldots\}$, we define the language[5] as follows:

$$\mathcal{L} \quad := \quad p_i \big| \neg A \big| A \wedge B \big| A \vee B \big| A \rightarrow B$$

We also use the logical constant $\curlywedge$ (*falsum*), denoting an absurd proposition.

An informational semantics is proposed in [10, 12] with a set of truth-values $\{1, 0, \bot\}$ reading: 1 = *known as true*, 0 = *known as false* and $\bot$ = *unknown*. A partial order (or *poset*, Definition A.1) of information $(\{1, 0, \bot\}, \leq)$ is depicted in Figure 2. The truth-tables for boolean connectives in Figure 3 coincide with the classical truth-tables over $\{1, 0\}$ and expand Kleene's 3-valued semantics [32] making it non-deterministic, e.g. $\bot \vee \bot \in \{1, \bot\} = f_\vee(\bot, \bot)$. Under this reading, A and B can be unknown $(A, B \mapsto \bot)$ but their disjunction known $A \vee B \mapsto 1$ (or the conjunction known to be false).[6]

[5] Let us note that a valuation (Definition 1.2) might assign different truth-values to classically equivalent formulas such as $\{A \rightarrow B, \neg A \vee B\}$. This is not the case if $\mathcal{L}$ is defined from minimally functionally complete sets, e.g. $\{\neg, \vee\}$ defining $A \rightarrow B := \neg A \vee B$ and so on.

[6] The truth-tables in Kleene's 3-valued logic are as in Figure 3 except for: $\bot \vee \bot = \bot \wedge \bot = \bot \rightarrow \bot = \bot$. Quine [38] motivates the informational reading of $\vee$ and $\wedge$ with the examples *this is a mouse or a chipmunk* and *this is a mouse and a chipmunk*.

Definition 1.2. (*Valuation*) A *valuation* over $\mathcal{L}$ is a function $V : \mathcal{L} \to \{1, 0, \bot\}$ satisfying the truth-tables in Figure 3.

The constant function $V_\bot : \mathcal{L} \to \{\bot\}$ satisfies the truth-tables, and so is a (null information) valuation. The set of designated values is just $\{1\}$, from which the 0-depth logic $(\mathcal{L}, \models_0)$ is defined as usual.

Definition 1.3. ($\models_0$ *consequence*) Given $\Gamma \cup \{A\} \subseteq \mathcal{L}$, we say that A is a *0-depth consequence* of Γ, denoted $\Gamma \models_0 A$, if for any valuation V, $V[\Gamma] = \{1\}$ implies $V(A) = 1$. Γ is 0-*depth consistent* if there is a valuation V with $V[\Gamma] = 1$. (Note we make use of the notation $f[X] = \{f(x) : x \in X\}$.)

The logic $\models_0$ is a Tarskian consequence relation, as it satisfies reflexivity, monotonicity and cut [11]. $\models_0$ is also structural but has no tautologies. Classical inconsistencies are assigned $\bot$ by $V_\bot$, 0 by any two-valued valuation $V : \mathcal{L} \to \{1, 0\}$, and even 1 if 0-depth consistent. Still, 0-depth inconsistent sets are explosive: $\{p, \neg p\} \models_0 \curlywedge$.

Example 1. (*Comparison with classical logic*) In 0-depth reasoning:

- the deduction theorem fails: $\{p\} \models_0 p$ but $\not\models_0 p \to p$;
- the commutative rule for $\vee$ can fail $\{p \vee q\} \not\models_0 q \vee p$; and also *de Morgan* $\{\neg(p \wedge q)\} \not\models_0 \neg p \vee \neg q$ and *full resolution* $\{p \vee q, \neg p \vee q\} \not\models_0 q$;
- some classical inconsistencies can take the value 1, e.g. $\{p \to \neg p, \neg p \to p\}$ or the CNF formula $\neg((p \wedge q) \vee (p \wedge \neg q) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q))$.

Definition 1.4. ($\vdash_0$-*proof*) A $\vdash_0$-*proof* of A from Γ, denoted $\Gamma \vdash_0 A$ if it exists, is a tree (N, R) of nodes labelled[7] with formulas $N \subseteq \mathcal{L}$ and such that: (i) A is the root node and leaf nodes are in Γ; (ii) if $A_0 \in N$ has as R-successors A_1, A_2 (possibly with $A_1 = A_2$), then

$$\frac{A_1 \quad A_2}{A_0} \text{ is an instance of an intelim rule (Fig. 4).}$$

Proposition 1.5. ($\vdash_0$ soundness; completeness [10, Proposition 4.7]) *The calculus* $\vdash_0$ *is sound and complete w.r.t. logical consequence* $\models_0$.

The logic of k-depth $\models_k$ strengthens $\models_0$ with at most k nested applications of *bivalence*. Under the informational semantics, each application of bivalence expands the information possessed by the agent with the virtual (or temporary) possession of some formula A, and in parallel of its negation $\neg A$, and extracts their shared consequences.[8]

[7] When no confusion exists, definitions and proofs omit the node/label distinction.

[8] Recall that in classical semantics all formulas are assumed bivalent from start.

introduction:

$$\frac{A \quad B}{A \wedge B} \qquad \frac{\neg A}{\neg(A \wedge B)} \qquad \frac{\neg B}{\neg(A \wedge B)} \qquad \frac{A}{\neg\neg A}$$

$$\frac{\neg A \quad \neg B}{\neg(A \vee B)} \qquad \frac{A}{A \vee B} \qquad \frac{B}{A \vee B} \qquad \frac{A \quad \neg A}{\curlywedge}$$

$$\frac{A \quad \neg B}{\neg(A \to B)} \qquad \frac{\neg A}{A \to B} \qquad \frac{B}{A \to B}$$

elimination:

$$\frac{\neg(A \wedge B) \quad A}{\neg B} \qquad \frac{\neg(A \wedge B) \quad B}{\neg A} \qquad \frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B} \qquad \frac{\neg\neg A}{A}$$

$$\frac{A \vee B \quad \neg A}{B} \qquad \frac{A \vee B \quad \neg B}{A} \qquad \frac{\neg(A \vee B)}{\neg A} \qquad \frac{\neg(A \vee B)}{\neg B} \qquad \frac{\curlywedge}{A}$$

$$\frac{A \to B \quad A}{B} \qquad \frac{A \to B \quad \neg B}{\neg A} \qquad \frac{\neg(A \to B)}{A} \qquad \frac{\neg(A \to B)}{\neg B}$$

Figure 4. The natural deduction system $\vdash_0$ consists of rules $I*, E*$ (called *intelim* rules) for the introduction and elimination of connectives and falsum $* \in \{\wedge, \vee, \to, \neg, \curlywedge\}$. D'Agostino [10] presents these rules in a language with signed formulas $\{TA, FA\}$, where they enjoy the *separation property* [4] so any rule for $*$ only mentions this connective $*$

$$\dfrac{\begin{array}{cc}\{A\}^1 & \{\neg A\}^2 \\ \vdots & \vdots \\ B & B\end{array}}{B}\;{}^{1,2} \qquad\qquad \dfrac{\begin{array}{cc}\Gamma, \{A\}^1 & \Delta, \{\neg A\}^2 \\ \vdots\, \Pi_1 & \vdots\, \Pi_2 \\ B & B\end{array}}{B}\;{}^{1,2}$$

Figure 5. (Left) A schematic application of the RB rule. A is called the RB-formula. Number labels are also used to track which assumptions are discharged at each RB application. (Right) Given a proof Π_1 of B from A and a proof Π_2 of B from $\neg A$, an application of RB results in a proof of B from $\Gamma \cup \Delta$. It combines Π_1, Π_2 and a root node labelled B

Definition 1.6. ($\models_k$ *consequence*) Define $\models_0$ as in Definition 1.3, and inductively:

$$\Gamma \models_{k+1} B \text{ iff } \text{ there is } A \in \mathcal{L} \text{ such that } \Gamma \cup \{A\} \models_k B \text{ and } \Gamma \cup \{\neg A\} \models_k B.$$

Each consequence relation $\models_k$ is monotonic and satisfies a weaker version of cut (see Proposition 6.6). For the corresponding system $\vdash_k$, one simply needs to add the rule of bivalence RB (Figure 5) and limit its nesting in proofs.

Definition 1.7. ($\vdash_k$*-proof*) A $\vdash_k$*-proof* of A from Γ is a tree (N, R) as in Definition 1.4 satisfying conditions (i)–(ii) with rules from Figures 4 and 5, and also:

(iii) each branch of the tree traverses $\leq k$ applications of RB.

Example 2. (RB and depth) Since $\{p\} \vdash_0 p \vee \neg p$ and $\{\neg p\} \vdash_0 p \vee \neg p$, an application of RB with the (discharged) pair $\{p, \neg p\}$ gives $\emptyset \vdash_1 p \vee \neg p$. Note that the number of classical theorems provable in $\vdash_k$ increases with k.

An important property of the DBBL hierarchy is that to decide whether $\Gamma \vdash B$, the RB-formulas can be searched among sets of bounded size such as the sets of subformulas $\mathsf{sub}(\Gamma \cup \{B\})$ or atoms $\mathsf{at}(\Gamma \cup \{B\})$.

Definition 1.8. (*Virtual space function*; $\models^{\mathsf{f}}_k$, $\vdash^{\mathsf{f}}_k$) A *virtual space function* is a function $\mathsf{f} : \mathcal{P}_{\mathsf{fin}}(\mathcal{L}) \to \mathcal{P}(\mathcal{L})$ satisfying: (i) $\mathsf{at}(\Delta) \subseteq \mathsf{f}(\Delta)$, (ii) $\mathsf{f}(\Delta)$ is closed under subformulas, and (iii) the size of $\mathsf{f}(\Delta)$ is polynomially bounded w.r.t. Δ. Any such function f defines a consequence $\models^{\mathsf{f}}_k$ and calculus $\vdash^{\mathsf{f}}_k$ as follows. Let $\models^{\mathsf{f}}_0 = \models_0$ resp. $\vdash^{\mathsf{f}}_0 = \vdash_0$ and:

$\Gamma \models^{\mathsf{f}}_{k+1} B$ iff there is $A \in \mathsf{f}(\Gamma \cup \{B\})$ with $\Gamma \cup \{A\} \models^{\mathsf{f}}_k B$ and $\Gamma \cup \{\neg A\} \models^{\mathsf{f}}_k B$;

$\Gamma \vdash^{\mathsf{f}}_{k+1} B$ iff $\Gamma \vdash_{k+1} B$ holds with RB-formulas satisfying $A \in \mathsf{f}(\Gamma \cup \{B\})$.[9]

The *length* of a formula A, denoted $|A|$, is the number of symbol occurrences in A. For a finite set $\Gamma \subseteq \mathcal{L}$, we define $|\Gamma| = \Sigma_{A \in \Gamma} |A|$.

Proposition 1.9. (Approximation; [10, Prop. 3.2]) *For any virtual space function* f*, it holds that* $\models \, = \, \bigcup_k \models^{\mathsf{f}}_k$.

Proposition 1.10. (DBBL tractability [11, Proposition 3.2]) *Let* $n = |\Gamma \cup \{A\}|$
be the length of a finite set $\Gamma \cup \{A\}$*. For any virtual space function* f*, the complexity of deciding whether* $\Gamma \vdash^{\mathsf{f}}_k A$ *is* $O(n^{k+2})$.

Proposition 1.11. ($\vdash^{\mathsf{f}}_k$ soundness; completeness; [10, Prop. 4.11]) *For any function* f *as in Definition 1.8 and* $k < \omega$, $\Gamma \models^{\mathsf{f}}_k A$ *iff* $\Gamma \vdash^{\mathsf{f}}_k A$.

2. Measures of Proof Complexity

A quick look at Definition 1.6 shows that k-depth proofs contain a branch with k nested applications of RB while other branches may differ from this

[9] Proofs based on *analytic cuts* (that is, $\mathsf{f} = \mathsf{sub}$ for RB) can be exponentially longer than those based on *unrestricted cut*, as witnessed by pigeon-hole formulas [6]. Analytic cuts, on the other hand, enjoy the *weak subformula property* [13] and reduce the search space in implementations by controling the choice of RB-formulas. In Section 7, we return to the general case and describe a modal logic based on unrestricted cuts.

Table 1. (Top, bottom) Examples where degree $h(\Gamma, B)$ matches minimum depth k. (Mid) An example where they do not match: $k < h(\Gamma, B)$. In each case, a $\vdash^{\mathsf{f}}_k$-proof of B obtains from applying RB to all pairs $\{p_i, \neg p_i\}$ occurrying in Γ, and the interim rules $\{E{\to}, I{\wedge}\}$

Γ	$\Gamma \models_k B$	$h(\Gamma, B)$
$\left\{\begin{array}{c} \pm p_1 \to q, \\ q \wedge \pm p_2 \to B \end{array}\right\}$	$2 \leq k$	2
$\left\{\begin{array}{cc} \pm p_1 \to q_1, & \pm p_2 \to q_2, \\ q_1 \wedge p_3 \to B, & q_2 \wedge \neg p_3 \to B \end{array}\right\}$	$2 \leq k$	3
$\left\{\begin{array}{r} \pm p_1 \to q_1, q_1 \wedge \pm p_2 \to q_2, \\ q_2 \wedge \pm p_3 \to B \end{array}\right\}$	$3 \leq k$	3

number. Another measure of proof complexity would then simply be:

$$h(\Gamma, B) = \text{the minimum number of formula pairs } \{A, \neg A\} \text{ needed as } RB\text{-assumptions to show that } \Gamma \models B.$$

Remark 1. (*Notation*) (± *over formulas.*) From here on, $\pm A$ denotes a choice between A and its negation, i.e. $\pm A \in \{A, \neg A\}$. $\{\pm A_1, \ldots, \pm A_k\}$ denotes a set obtained from independent choices for $\pm A_1, \ldots, \pm A_k$.

(± *inside formulas.*) In examples, ± is used for succintness: e.g. a set $\{\pm p \to q, \ldots\}$ denotes the set with *both* formulas $\{p \to q, \neg p \to q, \ldots\}$.

Definition 2.1. (*Degree*) The degree of (Γ, B), denoted $h(\Gamma, B)$, is the minimum $n \in \mathbb{N}$ such that there exist $A_1, \ldots, A_n$ with $\Gamma \cup \{\pm A_1, \ldots, \pm A_n\} \models_0 B$.

Example 3. (*Depth vs. degree*) See Table 1 for a comparison between degree $h(\Gamma, B)$ and minimum depth k. In Table 1 (mid), depth 2 corresponds to degree 3: each branch from $\pm p_3$ chooses a different RB pair ($\pm p_1$ resp. $\pm p_2$).

If k is minimal with $\Gamma \models_k B$, one can only narrow down the degree $h(\Gamma, B)$ by: $k \leq h(\Gamma, B) \leq 2^k - 1$.

Proposition 2.2. (Depth and degree) *The following implications hold:*

$$h(\Gamma, B) \leq k \quad \Rightarrow \quad \Gamma \models_k B \quad \Rightarrow \quad h(\Gamma, B) \leq 2^k - 1.$$

Proof. By induction on k. (Base Case $k = 0$) For an empty set of virtual assumptions, we obtain $0 = 2^0 - 1$. (Ind. Case $k \mapsto k+1$.) Assume as inductive hypothesis (IH) that the claim holds for k. (⇒1) Suppose that $h(\Gamma, B) \leq k + 1$. All cases where $h(\Gamma, B) < k+1$ follow from the IH plus monotonicity $\models_i \subseteq \models_{i+1}$, so let $h(\Gamma, B) = k+1$. By definition, there are $A_1, \ldots, A_{k+1}$ such

that $\Gamma \cup \{\pm A_1, \ldots, \pm A_{k+1}\} \models_0 B$. Applying the IH gives $\Gamma \cup \{A_{k+1}\} \models_k B$ and $\Gamma \cup \{\neg A_{k+1}\} \models_k B$, so Definition 1.6 gives $\Gamma \models_{k+1} B$.
($\Rightarrow$2) Now assume $\Gamma \models_{k+1} B$, so $\Gamma \cup \{A\} \models_k B$ and $\Gamma \cup \{\neg A\} \models_k B$, for some A. Applying twice the IH, we obtain two sets of formulas such that

$$(\Gamma \cup \{A\}) \cup \{\pm A_1, \ldots, \pm A_{2^{k+1}-1}\} \models_0 B \text{ and}$$
$$(\Gamma \cup \{\neg A\}) \cup \{\pm A'_1, \ldots, \pm A'_{2^{k+1}-1}\} \models_0 B.$$

Thus it also holds that $\Gamma \cup \{\pm A_1, \ldots, \pm A_{2^k-1}, \pm A'_1, \ldots, \pm A'_{2^k-1}, \pm A\} \models_0 B$. Or, after renaming the formulas, $\Gamma \cup \{\pm A_1, \ldots, \pm A_{2\cdot(2^k-1)+1}\} \models_0 B$. Since $2 \cdot (2^k - 1) + 1 = 2^{k+1} - 2 + 1 = 2^{k+1} - 1$ we are done. ■

A hierarchy of degree-bounded logics $\langle \Vdash_k \rangle_{k<\omega}$ satisfying $\Gamma \Vdash_k B$ iff $h(\Gamma, B) \leq k$, will be defined (Definition 6.1) using updates on valuations.

3. Preliminaries: a Complete Lattice of Valuations

The lattice structure of valuations obtains by lifting the order $\leq$ from truth-values (Figure 2). A similar construction for partial valuations is found in [9].

Definition 3.1. (*Set of valuations over $\mathcal{L}$*) Let us denote by $\mathbb{V}$ the set of valuations over $\mathcal{L}$: $\mathbb{V} = \big\{V : \mathcal{L} \to \{1, 0, \perp\} \mid V \text{ is a valuation}\big\}$.

Proposition 3.2. (Characterization; Prop. [11, 2.26]) *The set $\mathbb{V}$ of valuations is exactly the set of functions V such that (i) for no pair $A, \neg A$ it holds that $V(A) = 1 = V(\neg A)$ and (ii) V is closed under intelim rules.*

Definition 3.3. (*Information order* [11]) Given two valuations V, V', we say that V' *refines* V, denoted $V \sqsubseteq V'$, if for every $A \in \mathcal{L}$, $V(A) \leq V'(A)$. In other words, $V \sqsubseteq V'$ iff $V(A) = \perp$ or $V'(A) = V(A)$.

For a poset $(X, \leq)$ and a set $Y \subseteq X$, the *meet* $\bigsqcap Y$ is the greatest lower bound of Y (Definition A.2). The *join* $\bigsqcup Y$ is the least upper bound of Y. We also use the notation $x \sqcap y = \bigsqcap\{x, y\}$ and $x \sqcup y = \bigsqcup\{x, y\}$.

Example 4. (*Meet and join*) For the poset $(\{1, 0, \perp\}, \leq)$, its meets are: $\bigsqcap\{v\}$
$= v$ and otherwise $\bigsqcap X = \perp$; joins are $\bigsqcup\{v\} = v = \bigsqcup\{v, \perp\}$ while $\bigsqcup\{1, 0\}$, $\bigsqcup\{1, 0, \perp\}$ do not exist. The poset $(\mathbb{V}, \sqsubseteq)$ has no maximum (Figure 6) but is closed under meets [11] and so it is a meet semilattice (Definition A.3).

Proposition 3.4. *For any non-empty subset $\mathbb{V}' \subseteq \mathbb{V}$, the meet exists in $\mathbb{V}$ as the pointwise meet $\bigsqcap \mathbb{V}' = \{\langle A, \bigsqcap_{V \in \mathbb{V}'} V(A)\rangle : A \in \mathcal{L}\}$.*

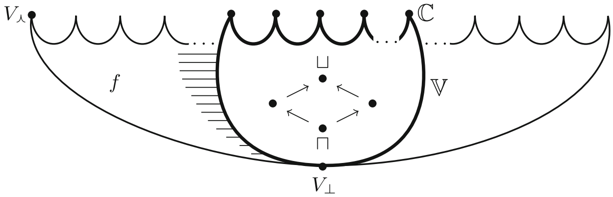

Figure 6. The information order $\sqsubseteq$ on the set of 3-valued *functions* $f : \mathcal{L} \to \{1, 0, \perp\}$. Its maximal elements are the 2-valued functions $f : \mathcal{L} \to \{1, 0\}$, including the constant function $V_\curlywedge : \mathcal{L} \to \{1\}$ and the set $\mathbb{C}$ of classical valuations. The $\sqsubseteq$-minimum element is $V_\perp : \mathcal{L} \to \{\perp\}$. $(\mathbb{V}, \sqsubseteq)$ is a meet semilattice; any *valuation* either satisfies a classical inconsistency (striped area) or it can be refined into a classical valuation (thick line)

Lemma 3.5. (Complete lattices [16, 2.30]) $(X, \leq)$ *is a complete lattice (Definition A.3) iff* (i) $(X, \leq)$ *contains a top element and* (ii) *it is closed under meets of non-empty sets. In the latter case, the join is defined by* $\bigsqcup Y = \bigsqcap Y^u$. *(That is, the meet of the set* Y^u *of upper bounds of* Y*, Definition A.2.)*

The set $\mathbb{V}$ is closed under meets (Proposition 3.4) but lacks a top element that acts as the join $V \sqcup V'$ of incompatible valuations, i.e. $V(A) = 1$ and $V'(A) = 0$. In [9] a complete lattice results from adding an inconsistent "valuation" as top. Our top element, $V_\curlywedge$, similarly makes all formulas true.

Definition 3.6. (*Poset of valuations with top*) Let $V_\curlywedge$ denote the constant function $V_\curlywedge : \mathcal{L} \to \{1\}$. We define an expansion of $(\mathbb{V}, \sqsubseteq)$ into $(\mathbb{V}^+, \sqsubseteq^+)$ by: $\mathbb{V}^+ = \mathbb{V} \cup \{V_\curlywedge\}$ and $V \sqsubseteq^+ V' \Leftrightarrow V \sqsubseteq V'$ or $V' = V_\curlywedge$.

Proposition 3.7. *The pair* $(\mathbb{V}^+, \sqsubseteq^+)$ *is a complete lattice. The join of any set* $\mathbb{V}' \subseteq \mathbb{V} \cup \{V_\curlywedge\}$ *is then* $\bigsqcup \mathbb{V}' = \bigsqcap (\mathbb{V}')^u$.

Proof. We check the two conditions from Lemma 3.5. (i) The top element is $V_\curlywedge$ by definition of $\sqsubseteq^+$. (ii) The meet extends to all new cases $\mathbb{V}' \cup \{V_\curlywedge\} \subseteq \mathbb{V}^+$: if $\mathbb{V}' = \emptyset$, then $\bigsqcap \{V_\curlywedge\} = V_\curlywedge$; and otherwise $\bigsqcap (\mathbb{V}' \cup \{V_\curlywedge\}) = \bigsqcap \mathbb{V}'$.[10] ■

To keep the terminology simple, we henceforth rename $\sqsubseteq^+$ as $\sqsubseteq$ and call any element of $\mathbb{V}^+$ a *valuation*. Note that although $V_\curlywedge$ does not satisfy the truth-tables and so $V_\curlywedge \notin \mathbb{V}$, its choice as a top element is not arbitrary.

Fact 3.8. $\mathbb{V}^+$ *is the set of functions* $V : \mathcal{L} \to \{1, 0, \perp\}$ *closed under applications of intelim rules.*

[10] Thus, the truth-value of B in $\bigsqcap (\mathbb{V}' \cup \{V_\curlywedge\})$ is no longer its pointwise meet, but the pointwise meet of $\bigsqcap \mathbb{V}'$. For an illustration, the pointwise meet of $\{V_\curlywedge, V_{\{\neg p\}}\}$ would give $V_\curlywedge(p) \sqcap V_{\{\neg p\}}(p) = 1 \sqcap 0 = \perp$ instead of $(V_\curlywedge \sqcap V_{\{\neg p\}})(p) = V_{\{\neg p\}}(p) = 0$.

4. A bit of Model Theory for 0-depth Logic

One can first observe that valuations and 0-depth theories are closely connected through the corresponding partial orders $\sqsubseteq$ and $\subseteq$.

Definition 4.1. (*Theory*) The set of 0-depth *consequences* of Γ is $\mathsf{Cn}_0(\Gamma) = \{A \in \mathcal{L} : \Gamma \vdash_0 A\}$. A set $\Gamma \subseteq \mathcal{L}$ is a *theory* if $\mathsf{Cn}_0(\Gamma) \subseteq \Gamma$. Abusively, we define the *theory* of a valuation V as the set $\mathsf{Th}(V) = \{A \in \mathcal{L} : V(A) = 1\}$.

Fact 4.2. *For any $V \in \mathbb{V}^+$, $\mathsf{Th}(V)$ is a theory: $\mathsf{Cn}_0(\mathsf{Th}(V)) \subseteq \mathsf{Th}(V)$.*

Lemma 4.3. (Orders) *For any $V, V' \in \mathbb{V}^+$, $\mathsf{Th}(V) \subseteq \mathsf{Th}(V')$ iff $V \sqsubseteq V'$.*

An interesting fact about 0-depth logic $\models_0$ is that among all the valuations satisfying a set of formulas Γ, one can construct their minimum. This minimum captures what all these valuations agree upon.

Definition 4.4. (*Set of valuations of* Γ) For any set $\Gamma \subseteq \mathcal{L}$, we define $\mathbb{V}_\Gamma$ as the set of valuations in $\mathbb{V}^+$ satisfying Γ, i.e. $V \in \mathbb{V}_\Gamma$ iff $V[\Gamma] = \{1\}$.

Whenever Γ is 0-depth consistent, $\mathbb{V}_\Gamma$ will contain some valuation in $\mathbb{V}$. All 0-depth inconsistent sets Γ share the same set of valuations $\mathbb{V}_\Gamma = \{V_\curlywedge\}$. In both cases, the infimum $\bigsqcap \mathbb{V}_\Gamma$ (Proposition 3.4) happens to be in $\mathbb{V}_\Gamma$ and so it is a minimum: $V_\Gamma = \bigsqcap \mathbb{V}_\Gamma$. A stepwise version of a construction used in the completeness proof [10, Proposition 4.7] gives us a direct proof of this fact.

Proposition 4.5. *For each set $\Gamma \subseteq \mathcal{L}$, a $\sqsubseteq$-minimum valuation, called V_Γ, exists in $\mathbb{V}_\Gamma$. As a consequence $V_\Gamma = \bigsqcap \mathbb{V}_\Gamma$.*

Definition 4.6. (*Up-set*) Given a poset $(X, \leq)$, the *up-set* of an element $y \in X$ is $\uparrow y = \{x \in X : y \leq x\}$.

From Propositions 3.4–4.5 it follows that model-checking V_Γ suffices for checking any $\models_0$-consequence A of Γ, offering an insight into the low complexity of its validity problem, namely $O(n^2)$ for $n = |\Gamma \cup \{A\}|$. It also follows that a 1-1 correspondence exists between theories and minimum valuations.

Fact 4.7. *For any set $\Gamma \subseteq \mathcal{L}$, (i) $\mathbb{V}_\Gamma = \uparrow V_\Gamma$ and (ii) $\bigsqcap \uparrow V_\Gamma = V_\Gamma$.*

Corollary 4.8. *For any set $\Gamma \subseteq \mathcal{L}$, it holds that $\Gamma \models_0 A$ iff $V_\Gamma(A) = 1$.*

Proposition 4.9. (Correspondence) (i) *For any theory Γ, it holds that $\Gamma = \mathsf{Th}(V_\Gamma)$.* (ii) *For any $V' \in \mathbb{V}$, it holds that $V' = V_{\mathsf{Th}(V')}$.*

Proof. We use that Γ is a theory plus Proposition 1.5 and Corollary 4.8 to obtain:

(i) $A \in \Gamma$ iff $\Gamma \vdash_0 A$ iff $\Gamma \models_0 A$ iff $V_\Gamma(A) = 1$ iff $A \in \mathsf{Th}(V_\Gamma)$;

(ii) $V'(A) = 1$ iff $A \in \mathsf{Th}(V')$ iff $\mathsf{Th}(V') \models_0 A$ iff $V_{\mathsf{Th}(V')}(A) = 1$

which gives $\mathsf{Th}(V') = \mathsf{Th}(V_{\mathsf{Th}(V')})$. By Lemma 4.3, $V' = V_{\mathsf{Th}(V')}$. ■

Corollary 4.10. *For any* $\Gamma \subseteq \mathcal{L}$, (i) $V_\Gamma = V_{\mathsf{Cn}_0(\Gamma)}$ *and* (ii) $\mathsf{Cn}_0(\Gamma) = \mathsf{Th}(V_\Gamma)$.

Fact 4.11. $\mathsf{Th}(V_1 \sqcap V_2) = \mathsf{Th}(V_1) \cap \mathsf{Th}(V_2)$. *As a consequence,* $V_\Gamma \sqcap V_\Delta = V_{\mathsf{Cn}_0(\Gamma) \cap \mathsf{Cn}_0(\Delta)}$.

Proposition 4.12. (Join operator in $\mathbb{V}^+$) *The join* $\bigsqcup \mathbb{V}' = \bigsqcap (\mathbb{V}')^u$ *obtained from Proposition* 3.7 *is:* $\bigsqcup \mathbb{V}' = V_{\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')}$. *Hence,* $V_\Gamma \sqcup V_\Delta = V_{\Gamma \cup \Delta}$.

The connectives $\wedge, \vee$ need not be distributive over each other, as witnessed by valuations V satisfying one of the following:

$$V(\bot \wedge (1 \vee 1)) = \bot < 1 = V((\bot \wedge 1) \vee (\bot \wedge 1)) \in \{1, \bot\}$$
$$\{\bot, 0\} \ni V(0 \vee (\bot \wedge \bot)) = \bot < 0 = V((0 \vee \bot) \wedge (0 \vee \bot)) \in \{0, \bot\}.$$

In a similar vein, the lattice of valuations $(\mathbb{V}^+, \sqsubseteq)$ is semi-distributive (see Lemma A.8) but not distributive as shown in Example 5 below.

Corollary 4.13. (Join operator, cont'd) $\bigsqcup \mathbb{V}'$ *is either* $V_\curlywedge$ *or the pointwise join* $\{\langle A, \bigsqcup_{V \in \mathbb{V}'} V(A) \rangle : A \in \mathcal{L}\}$.

Proof. Suppose first that $V_\curlywedge \in \mathbb{V}'$ or that $V(A) = 1$ and $V'(A) = 0$ for some $V, V' \in \mathbb{V}'$ (recall that $1 \sqcup 0$ is undefined). In either case there are $V, V' \in \mathbb{V}$ with $V(A) = 1 = V(\neg A)$ and so $\bigcup_{V \in \mathbb{V}'} \mathsf{Th}(V) = \mathcal{L}$. Then, Lemma 4.12 gives $\bigsqcup \mathbb{V}' = V_\curlywedge$.

Suppose now the opposite: $V_\curlywedge \notin \mathbb{V}'$ and $\{V(A)\}_{V \in \mathbb{V}'} \notin \{\{1, 0\}, \{1, 0, \bot\}\}$ for each A. Hence each such set $\{V(A)\}_{V \in \mathbb{V}'}$ has a join. Moreover,

$$(\bigsqcup \mathbb{V}')(A) = 1 \quad \text{iff} \quad A \in \bigcup_{V \in \mathbb{V}'} \mathsf{Th}(V) \quad \text{iff} \quad (\bigsqcup_{V \in \mathbb{V}'} V(A)) = 1.$$

Equivalently put, $\mathsf{Th}(\bigsqcup \mathbb{V}') = \mathsf{Th}(\{\langle A, \bigsqcup_{V \in \mathbb{V}'} V(A) \rangle : A \in \mathcal{L}\})$. One can then apply Lemma 4.3 to conclude $\bigsqcup \mathbb{V}' = \{\langle A, \bigsqcup_{V \in \mathbb{V}'} V(A) \rangle : A \in \mathcal{L}\}$. ■

5. Updates and the *RB*-Rule

Intuitively, updating a valuation V with a formula A should result in a valuation that refines V, satisfies A and is minimal with these properties.

Definition 5.1. (A-*refinement* [12], A-*update*) Given $V \in \mathbb{V}^+$ and $A \in \mathcal{L}$, we say that $V' \in \mathbb{V}^+$ is a A-*refinement* of V if: (i) $V \sqsubseteq V'$ and (ii) $V'(A) = 1$. The A-*update* of V is the $\sqsubseteq$-minimum A-refinement of V, denoted V^A.

Proposition 5.2. *(A-update) The A-update of V is $V^A = V \sqcup V_{\{A\}}$.*

Proof. Conditions (i) and (ii) in Definition 5.1 give that the set of A-refinements of V is $\uparrow V \cap \mathbb{V}_{\{A\}}$. By Fact 4.7, this set is $\uparrow V \cap \uparrow V_{\{A\}}$ and by Lemma A.4, it is $\uparrow (V \sqcup V_{\{A\}})$. Hence the join $V \sqcup V_{\{A\}}$ is the least upper bound of the A-refinements of V. Since $V, V_{\{A\}} \sqsubseteq V \sqcup V_{\{A\}}$, $V \sqcup V_{\{A\}}$ is also in the set of such refinements, and so it is the $\sqsubseteq$-minimum A-refinement of V. ■

Corollary 5.3. (Trivial updates) *(i) If $V(A) = 1$, then $V^A = V$; (ii) if $V(A) = 0$, then $V^A = V_{\curlywedge}$.*

Lemma 5.4. (Update as expansion) *For any $\Gamma \cup \{A\} \subseteq \mathcal{L}$, (i) $V_\Gamma^A = V_{\Gamma \cup \{A\}}$ and (ii) $(\mathbb{V}_\Gamma)^A = \mathbb{V}_{\Gamma \cup \{A\}}$, where $(\mathbb{V}_\Gamma)^A = \{V^A : V \in \mathbb{V}_\Gamma\}$.*

Proof. (i) From $V_\Gamma^A = V_\Gamma \sqcup V_{\{A\}}$ (Proposition 5.2), the claim is immediate from Proposition 4.12. (ii) If $V' \in \mathbb{V}_{\Gamma \cup \{A\}}$, then $V' \in \mathbb{V}_\Gamma$ and $V'(A) = 1$ so by Corollary 5.3 $V' = V'^A$ and then $V' \in (\mathbb{V}_\Gamma)^A$. Hence, $\mathbb{V}_{\Gamma \cup \{A\}} \subseteq (\mathbb{V}_\Gamma)^A$. Let now $V'^A \in (\mathbb{V}_\Gamma)^A$ so that $V' \in \mathbb{V}_\Gamma$. If $V'(A) = 1$, then $V'^A = V' \in \mathbb{V}_{\Gamma \cup \{A\}}$; if $V'(A) = 0$, then $V'^A = V_{\curlywedge} \in \mathbb{V}_{\Gamma \cup \{A\}}$; and if $V'(A) = \bot$, then $V'^A = V' \sqcup V_{\{A\}}$ is in $\mathbb{V}_\Gamma$ (since $V'^A \sqsupseteq V'$) and also in $\mathbb{V}_{\{A\}}$ (since $V'^A \sqsupseteq V_{\{A\}}$). Definition 4.4 gives $\mathbb{V}_\Gamma \cap \mathbb{V}_{\{A\}} = \mathbb{V}_{\Gamma \cup \{A\}}$, and so $V'^A \in \mathbb{V}_{\Gamma \cup \{A\}}$. This shows $(\mathbb{V}_\Gamma)^A \subseteq \mathbb{V}_{\Gamma \cup \{A\}}$. ■

Definition 5.5. ($\pm$-*update*) Given a formula $A \in \mathcal{L}$, the $[\pm]A$-*update* of a valuation V is defined as: $V^{[\pm]A} = V^A \sqcap V^{\neg A}$.

Example 5. ($\pm$-updates simulate *RB*) The following $\sqsubseteq$-chain below holds for any set Γ. The $\sqsubseteq$ inequalities can all be proper, as in $\Gamma = \{\pm p \to q\}$. For this Γ, each formula in the the lower line becomes true exactly at the valuation above it.

$$\begin{array}{ccccccc} V_\Gamma & \sqsubseteq & V_\Gamma^{p \vee \neg p} & \sqsubseteq & V_\Gamma^{[\pm]p} & \sqsubseteq & V_\Gamma^{p}, V_\Gamma^{\neg p} \\ \Gamma & & p \vee \neg p & & q & & p \quad \neg p. \end{array}$$

Each $\sqsubseteq$ inequality can also become an identity if the update is made trivial (Corollary 5.3 or Fact 5.6 below); e.g. for the first and third inequalities, the sets $\Gamma_1 = \{\pm p \to q, p \vee \neg p\}$ and resp. $\Gamma_3 = \{\pm p \to q, p\}$ give:

$$V_{\Gamma_1} = V_{\Gamma_1}^{p \vee \neg p} \qquad \text{and} \qquad V_{\Gamma_3}^{[\pm]p} = V_{\Gamma_3}^{p} (= V_{\Gamma_3}).$$

Example 6. ($(\mathbb{V}^+, \sqsubseteq)$ is not distributive) The inequality $V_\Gamma^{p \vee \neg p} \sqsubset V_\Gamma^{[\pm]p}$ seen above for $\Gamma = \{\pm p \to q\}$ shows that the lattice of valuations is not distributive. These two valuations are respectively:

$$V_\Gamma^{p\vee\neg p} = V_\Gamma \sqcup V_{\{p\vee\neg p\}} = V_\Gamma \sqcup (V_{\mathsf{Cn}_0(\{p\})\cap\mathsf{Cn}_0(\{\neg p\})}) = V_\Gamma \sqcup (V_{\{p\}} \sqcap V_{\{\neg p\}})^{11}$$
$$V_\Gamma^{[\pm]p} = V_\Gamma^{p} \sqcap V_\Gamma^{\neg p} = (V_\Gamma \sqcup V_{\{p\}}) \sqcap (V_\Gamma \sqcup V_{\{\neg p\}})$$

Given that $\langle q, \bot\rangle \in V_\Gamma^{p\vee\neg p}$ and $\langle q, 1\rangle \in V_\Gamma^{[\pm]p}$ distributivity fails since:

$$V_\Gamma \sqcup (V_{\{p\}} \sqcap V_{\{\neg p\}}) \sqsubsetneq (V_\Gamma \sqcup V_{\{p\}}) \sqcap (V_\Gamma \sqcup V_{\{\neg p\}}).$$

FACT 5.6. *(Trivial ±-updates) If $V(A) \neq \bot$, then $V^{[\pm]A} = V$.*

Any such valuation $V^{[\pm]A}$ enforces all (and only) the consequences of applying *RB* with the formula A. Since depth-bounded logics $\models_k$ admit different *RB*-formulas in each branch after an *RB* rule (for $k > 1$), depth and number of updates need not match beyond the $k = 1$ case shown below.

PROPOSITION 5.7. *For any virtual space function* f *(Definition 1.8),* $\Gamma \models_1^{\mathsf{f}} B$ *iff* $V_\Gamma^{[\pm]A}(B) = 1$ *for some* $A \in \mathsf{f}(\Gamma \cup \{B\})$.

EXAMPLE 7. (*Degree and ±-updates*) Recall the set from Table 1 (mid)

$$\Gamma = \{\pm p_1 \to q_1,\ \ \pm p_2 \to q_2,\ \ q_1 \wedge p_3 \to B,\ \ q_2 \wedge \neg p_3 \to B\}$$

satisfying $h(\Gamma, B) = 3$ and $\Gamma \models_2^{\mathsf{f}} B$, and let (j) denote the j-th premise from Γ. Three ±-updates (in any order, see Corollary 5.9 below) suffice for B:

$$(V_\Gamma^{[\pm]p_1})^{p_3} : q_1 \wedge p_3 \mapsto 1 \quad \text{by } (1) \qquad \text{and so } \ B \mapsto 1 \text{ by } (3)$$
$$(V_\Gamma^{[\pm]p_2})^{\neg p_3} : q_2 \wedge \neg p_3 \mapsto 1 \text{ by } (2) \qquad \text{and so } \ B \mapsto 1 \text{ by } (4).$$

Finally, $V_\Gamma^{[\pm]p_1[\pm]p_2[\pm]p_3}(B) = (V_\Gamma^{[\pm]p_1[\pm]p_2})^{p_3}(B) \sqcap (V_\Gamma^{[\pm]p_1[\pm]p_2})^{\neg p_3}(B) = 1.$

LEMMA 5.8. *(Sequential A-updates) For any valuation V, it holds that $(V^A)^B = (V^B)^A$ and also that $(V^A)^B = V^{A\wedge B}$.*

PROOF. For the first claim, use commutativity and associativity (Lemma A.7) and Proposition 5.2 and reason as follows:

$$(V^A)^B = (V \sqcup V_{\{A\}}) \sqcup V_{\{B\}} = (V \sqcup V_{\{B\}}) \sqcup V_{\{A\}} = (V^B)^A.$$

For the second claim, we use Corollary 4.10(i) and expand $(V^A)^B$ as follows:

$$(V \sqcup V_{\{A\}}) \sqcup V_{\{B\}} = V \sqcup (V_{\{A\}} \sqcup V_{\{B\}}) = V \sqcup V_{\{A,B\}}$$
$$= V \sqcup V_{\mathsf{Cn}_0(\{A,B\})} = V \sqcup V_{\mathsf{Cn}_0(\{A\wedge B\})} = V \sqcup V_{\{A\wedge B\}} = V^{A\wedge B}.$$

[11] Note that $V_{\{p\vee\neg p\}} = V_{\mathsf{Cn}_0(\{p\})} \sqcap V_{\mathsf{Cn}_0(\{\neg p\})}$ does not generalize into $V_{\{A\vee B\}} = V_{\mathsf{Cn}_0(\{A\})} \sqcap V_{\mathsf{Cn}_0(\{B\})}$, e.g. for $A = p = B$. The general property only holds if 0-depth logic is expanded with *idempotence* $A \vee A \vdash_0' A$, an option discussed in [12].

■

Corollary 5.9. (Permutation of ±-updates) *For any valuation $V \in \mathbb{V}^+$ and formulas $A, B \in \mathcal{L}$, $V^{[\pm]A\,[\pm]B} = V^{[\pm]B\,[\pm]A}$.*

Proof. We split the identity and prove both directions $\sqsubseteq$ and $\sqsupseteq$. For ($\sqsupseteq$),

$$\begin{aligned}
&(V^{[\pm]A})^{[\pm]B}\\
&\quad= (V^{[\pm]A})^{B} \sqcap (V^{[\pm]A})^{\neg B}\\
&\quad= (V^{[\pm]A} \sqcup V_{\{B\}}) \sqcap (V^{[\pm]A} \sqcup V_{\{\neg B\}})\\
&\quad= \Big((V^{A} \sqcap V^{\neg A}) \sqcup V_{\{B\}}\Big) \sqcap \Big((V^{A} \sqcap V^{\neg A}) \sqcup V_{\{\neg B\}}\Big)\\
&\quad\sqsupseteq (V^{A} \sqcup V_{\{B\}}) \sqcap (V^{\neg A} \sqcup V_{\{B\}}) \sqcap (V^{A} \sqcup V_{\{\neg B\}}) \sqcap (V^{\neg A} \sqcup V_{\{\neg B\}})\\
&\quad= (V^{A})^{B} \sqcap (V^{\neg A})^{B} \sqcap (V^{A})^{\neg B} \sqcap (V^{\neg A})^{\neg B}\\
&\quad= (V^{A})^{B} \sqcap (V^{A})^{\neg B} \sqcap (V^{\neg A})^{B} \sqcap (V^{\neg A})^{\neg B}\\
&\quad= (V^{B})^{A} \sqcap (V^{\neg B})^{A} \sqcap (V^{B})^{\neg A} \sqcap (V^{\neg B})^{\neg A}\\
&\quad= (V^{[\pm B]})^{A} \sqcap (V^{[\pm]B})^{\neg A} = (V^{[\pm B]})^{[\pm]A}
\end{aligned}$$

where each step follows from applications of: Definition 5.5; Proposition 5.2; Definition 5.5; ($\sqsupseteq$) Lemma A.8; Proposition 5.2; Lemma A.7 (commutativity); Lemma 5.8; and Definition 5.5 (twice). For ($\sqsubseteq$), switching A and B everywhere in the above reasoning gives $(V^{[\pm B]})^{[\pm]A} \sqsupseteq (V^{[\pm A]})^{[\pm]B}$, and so we are done. ■

6. UBBL: A Hierarchy of Logics of Bounded Degree

Let us finally define logics of bounded degree $\Vert\!\models_k$ via updates (Lemma 5.4) rather than expansions (Definition 2.1). Note that 0-degree logic is just $\Vert\!\models_0$ $= \models_0$.

Definition 6.1. (*k-degree consequence*) For any virtual space function f and set $\Gamma \cup \{B\} \subseteq \mathcal{L}$, define:

$\Gamma \Vert\!\models^{\mathsf{f}}_k B$ iff there are $A_1, \ldots, A_k \in \mathsf{f}(\Gamma \cup \{B\})$ such that $V^{[\pm]A_1 \cdots [\pm]A_k}(B) =$ 1 for all $V \in \mathbb{V}_\Gamma$.

Definition 6.2. ($\Vdash^{\mathsf{f}}_k$-*proof*) Given a virtual space function f, a $\Vdash^{\mathsf{f}}_k$-*proof* of B from Γ is a tree (N, R) satisfying (i)–(ii) from Definition 1.7 plus:

(iii') the tree contains $\leq k$ RB-formulas A_i with $A_1, \ldots, A_k \in \mathsf{f}(\Gamma \cup \{B\})$.

Lemma 6.3. (Soundness, completeness UBBL) *For any k, any function f and set $\Gamma \cup \{B\} \subseteq \mathcal{L}$, it holds that $\Gamma \Vdash^{\mathsf{f}}_k B$ iff $\Gamma \Vert\!\models^{\mathsf{f}}_k B$.*

Proof. (Base case 0) From definitions and Proposition 1.5, $\Vdash^{\mathsf{f}}_0 = \vdash_0 = \models_0 = \Vvdash^{\mathsf{f}}_0$. (Ind. case $k+1$.) Assume the claim holds for k and let $\Gamma \Vdash^{\mathsf{f}}_{k+1} B$. Given a $\Vdash^{\mathsf{f}}_{k+1}$-proof of B from Γ, one can split all RB-instances with, say, $\{A_1, \neg A_1\}$ and obtain two proofs witnessing $\Gamma \cup \{A_1\} \Vdash_k B$ resp. $\Gamma \cup \{\neg A_1\} \Vdash_k B$ whose RB-instances are all within a set $\{\pm A_2, \dots, \pm A_{k+1}\} \subseteq \mathsf{f}(\Gamma, B)$. And vice versa, any such pair of proofs can be merged so that all occurrences of $A_1, \neg A_1$ are discharged by new RB-instances. This justifies the first equivalence below. We omit everywhere *for some* $A_1, A_2, \dots A_{k+1} \in \mathsf{f}(\Gamma \cup \{B\})$:

$$
\begin{aligned}
& \Gamma \Vdash^{\mathsf{f}}_{k+1} B \\
\text{iff } & \Gamma \cup \{A_{k+1}\} \Vdash^{\mathsf{f}}_k B \ \text{ and } \Gamma \cup \{\neg A_{k+1}\} \Vdash^{\mathsf{f}}_k B \\
\text{iff } & V^{[\pm]A_1 \dots [\pm]A_k}(B) = 1 \text{ for all } V \in \mathbb{V}_{\Gamma \cup \{A_{k+1}\}} \cup \mathbb{V}_{\Gamma \cup \{\neg A_{k+1}\}} \\
\text{iff } & V^{[\pm]A_1 \dots [\pm]A_k}(B) = 1 \text{ for all } V \in (\mathbb{V}_\Gamma)^{A_{k+1}} \cup (\mathbb{V}_\Gamma)^{\neg A_{k+1}} \\
\text{iff } & (V^{[\pm]A_1 \dots [\pm]A_k})^{A_{k+1}}(B) = 1 \text{ for all } V \in \mathbb{V}_\Gamma \text{ and} \\
& \quad (V^{[\pm]A_1 \dots [\pm]A_k})^{\neg A_{k+1}}(B) = 1 \text{ for all } V \in \mathbb{V}_\Gamma \\
\text{iff } & (V^{[\pm]A_1 \dots [\pm]A_k})^{[\pm]A_{k+1}}(B) = 1 \text{ for all } V \in \mathbb{V}_\Gamma \\
\text{iff } & \Gamma \Vvdash^{\mathsf{f}}_{k+1} B
\end{aligned}
$$

where each step follows from: the reasoning above; the inductive hypothesis; Lemma 5.4; def. of $(\mathbb{V}_\Gamma)^A$; Definition 5.5 and meets; and resp. Definition 6.1. ■

Fact 6.4. *It holds that* $\Vvdash_1 = \models_1$ *and* $\Vvdash_k \subsetneq \models_k$ *for any* $k > 1$.

Proposition 6.5. (Approximation) $\models = \bigcup_k \Vvdash^{\mathsf{f}}_k$, *for any* f *as in Definition* 1.8.

Proposition 6.6. (Bounded transitivity) *For any* $\Gamma \cup \{A, B\} \subseteq \mathcal{L}$ *with* $A \in \mathsf{f}(\Gamma \cup \{B\})$, *if* $\Gamma \Vvdash^{\mathsf{f}}_i A$ *and* $\Gamma \cup \{A\} \Vvdash^{\mathsf{f}}_j B$ *then* $\Gamma \Vvdash^{\mathsf{f}}_{i+j} B$.[12]

Proposition 6.7. (`UBBL` tractability) *For a finite set* $\Gamma \cup \{A\} \subseteq \mathcal{L}$ *with* $n = |\Gamma \cup \{A\}|$ *and any virtual space function* f, *an upper bound for the complexity of the validity problem* $\Gamma \Vdash^{\mathsf{f}}_k A$ *is* $O(n^{k+2})$.

Proof. $\Gamma \Vdash^{\mathsf{f}}_k A$ implies $\Gamma \Vvdash^{\mathsf{f}}_k A$ (Lemma 6.3); which implies $\Gamma \models^{\mathsf{f}}_k A$ (Fact 6.4); and finally $\Gamma \vdash^{\mathsf{f}}_k A$ (Lemma 1.11). By Proposition 1.10, $O(n^{k+2})$ is an upper bound for any instance of $\Gamma \Vdash^{\mathsf{f}}_k A$. ■

Definition 6.8. (*Classical valuations*) Define $\mathbb{C} \subseteq \mathbb{V}$ as the set of classical valuations $\overline{V} : \mathcal{L} \to \{1, 0\}$, induced by atomic assignments $\overline{V} : \mathsf{Var} \to \{1, 0\}$.

[12] Note that the corresponding claim in [10] is missing the assumption $A \in \mathsf{f}(\Gamma \cup \{B\})$.

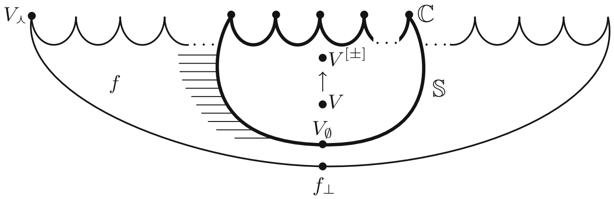

Figure 7. The set of 3-valued functions $f : \mathcal{L}_U \to \{1, 0, \perp\}$. (Top) $\sqsubseteq$-maximal elements are 2-valued, including the classical-like valuations in $\mathbb{C}$. (Bottom) the $\sqsubseteq$-minimum constant function $f_\perp : \mathcal{L}_U \to \{\perp\}$ is not a valuation over $\mathcal{L}_U$. The set $\mathbb{S}$ of all valuations (thick line and the striped area) is a U-model, whose $\sqsubseteq$-minimum $V_\emptyset$ is more informative than $f_\perp$. Functions in the striped area cannot be refined into $\mathbb{C}$

Lemma 6.9. *For any $V \in \mathbb{V}$, the following are equivalent:*

(i) $V \not\sqsubseteq \overline{V}$, *for any* $\overline{V} \in \mathbb{C}$*;*

(ii) $V^{[\pm]A_1 \cdots [\pm]A_k} = V_\curlywedge$, *for some* $A_1, \ldots, A_k \in \mathcal{L}$*;*

(iii) $V(A) = 1$ *for some classical inconsistency A, i.e.* $\vdash \neg A$.

Hence, for a set $\mathbb{V}'$ its join is the top element $\bigsqcup \mathbb{V}' = V_\curlywedge$ if $\mathbb{V}'$ contains either: $V_\curlywedge$, or a pair V, V' with $\{V(A), V'(A)\} = \{1, 0\}$ for some $A \in \mathcal{L}$, or finally $V, V_1, \ldots, V_k$ with V satisfying condition (ii) from Lemma 6.9 and $V_1(A_1) \neq \perp$, …, $V_k(A_k) \neq \perp$.

Example 8. (*Inconsistency*) The disjunction $(p_1 \wedge \neg p_1) \vee \ldots \vee (p_{k+2} \wedge \neg p_{k+2})$ is k-depth consistent and k+1-depth refutable. If a valuation V satisfies it, k+1 updates with e.g. $\{\pm p_1, \ldots, \pm p_{k+1}\}$ suffice to render the 0-depth inconsistent formula $p_{k+2} \wedge \neg p_{k+2}$ true, thus making $V^{[\pm]p_1 \cdots [\pm]p_{k+1}} = V_\curlywedge$.

7. A Modal Logic for *RB*-Updates

The characterization of the *RB* rule via updates not only defines degree-bounded propositional logics: it also provides a semantics for modalities that capture applications of *RB* and make the virtual assumptions explicit.

Definition 7.1. (*Language $\mathcal{L}_U$*) We expand the language $\mathcal{L}$ with an update operator $[\pm C]$ for every propositional formula $C \in \mathcal{L}$:

$$\mathcal{L}_U := p_i \mid \neg A \mid A \wedge B \mid A \vee B \mid A \to B \mid [\pm C]A$$

The information order $\sqsubseteq$ extends to functions f over $\mathcal{L}_U$. Valuations now depend on updates (Definition 7.2), making them relative to a model (Definition 7.3). See Figure 7 for an illustration and Definition 7.6 for an explicit construction.

Definition 7.2. (*Valuation over $\mathcal{L}_U$*) A *$\mathcal{L}_U$-valuation* is any function $V : \mathcal{L}_U \to \{1, 0, \bot\}$ respecting the truth-tables (Figure 3) and the *update condition*:

$$V([\pm A]B) = V^{[\pm]A}(B)$$

where $V^{[\pm]A} = V^A \sqcap V^{\neg A}$ and V^A is the $\sqsubseteq$-minimum valuation V' satisfying $V' \sqsupseteq V$ and $V'(A) = 1$. We also call the function $V_\curlywedge : \mathcal{L}_U \to \{1\}$ a valuation.

Definition 7.3. (*U-model*) A set $\mathbb{U}$ of valuations is a *U-model* iff it is closed under A-updates: $V \in \mathbb{U}$ implies $V^A \in \mathbb{U}$. For any set $\Gamma \cup \{A\} \subseteq \mathcal{L}_U$, define:

$$\begin{array}{ll}\Gamma \models_{\mathbb{U}} A \text{ iff } V[\Gamma] = \{1\} \text{ implies } V(A) = 1 & \text{for any } V \in \mathbb{U} \\ \Gamma \models_U A \text{ iff } \Gamma \models_{\mathbb{U}} A & \text{for any } U\text{-model } \mathbb{U}.\end{array}$$

Example 9. (*Valuations*) All $\mathcal{L}_U$-valuations V satisfy $V([\pm p]p \vee \neg p) = V^{[\pm]p}(p \vee \neg p) = 1$. Thus, their minimum $V_\emptyset$ (see Corollary 8.9) is not the null function $f_\bot : \mathcal{L}_U \to \{\bot\}$ (Figure 7), although $V_\emptyset$ does contain $V_\bot : \mathcal{L} \to \{\bot\}$.

Since U-models are closed under A-updates, they contain $V_\curlywedge$. We extend $\sqsubseteq$ as before to $V \sqsubseteq V_\curlywedge$ for any valuation V, so that $V \sqcap V_\curlywedge = V = V_\curlywedge \sqcap V$.

Remark 2. The meet of a set of $\mathcal{L}_U$ valuations $\bigsqcap \mathbb{S}'$ corresponds to the restricted pointwise meet, just as in the propositional case: $\bigsqcap\{V_\curlywedge\}(A) = 1$ and $(\bigsqcap \mathbb{S}')(A) = \bigsqcap_{V \in \mathbb{S}' \setminus \{V_\curlywedge\}} V(A)$. We observe that $V_{\{A\}}(\neg A) = 1$ implies $V_{\{A\}} = V_\curlywedge$ and that $V^{[\pm]A} = V_\curlywedge$ iff $V^A = V_\curlywedge = V^{\neg A}$. In fact, a valuation $V \neq V_\curlywedge$ can make a formula $[\pm A]B$ true through an update leading to $V^{[\pm]A} = V_\curlywedge$. For an example, if $B = \{(p \wedge \neg p) \vee (q \wedge \neg q)\}$ then $V_{\{B\}}^{[\pm]p} = V_\curlywedge$ and so $V_{\{B\}}([\pm p]B) = 1$, while clearly $V_{\{B\}} \neq V_\curlywedge$. (Compare with Lemma 6.9.)

Fact 7.4. *(Trivial and generated U-models) The set $\mathbb{U}_\curlywedge = \{V_\curlywedge\}$ is a U-model with $\models_{\mathbb{U}_\curlywedge} = \mathcal{P}(\mathcal{L}_U) \times \mathcal{L}_U$. If $\mathbb{U}$ is a U-model, then so is the submodel $\mathbb{U}_\Gamma = \{V \in \mathbb{U} : V[\Gamma] = 1\}$ generated by any set of formulas $\Gamma \subseteq \mathcal{L}_U$.*

In order to show that non-trivial U-models exist, we build one with all valuations through fragments $\mathcal{L}_{Uk}$ of the language $\mathcal{L}_U = \bigcup_{k<\omega} \mathcal{L}_{Uk}$.

Definition 7.5. ($\mathcal{L}_{Uk}, \mathcal{L}_{U\leq k}$ *fragments*) Let $\mathcal{L}_{U0} = \mathcal{L}$ and define:

$$\mathcal{L}_{Uk+1} := \quad [\pm A]B_i \quad \neg B \quad B \wedge C \quad B \vee C \quad B \to C.$$

where $A \in \mathcal{L}$ and $B_i \in \mathcal{L}_{U\leq k} := \bigcup_{i\leq k} \mathcal{L}_{Ui}$.

Definition 7.6. (*Standard model* $\mathbb{S}$) Define $\mathbb{S}_0 = \mathbb{V}$ and $\mathbb{S}_{n+1}$ as the set of functions V_{n+1} of the form:

$$\begin{aligned} V_{n+1} = V_n &\cup \{\langle [\pm B]C, (V_n^B \sqcap V_n^{\neg B})(C)\rangle : B \in \mathcal{L}, C \in \mathcal{L}_{Un}\} \\ &\cup \{\langle B * C, v\rangle : v \in f_*(V_n(B), V_n(C)), B * C \in \mathcal{L}_{Un+1} \setminus \mathcal{L}_{U\leq n}\} \\ &\cup \{\langle \neg B, f_\neg(V_n(B))\rangle : B \in \mathcal{L}_{Un+1} \setminus \mathcal{L}_{U\leq n}\} \end{aligned}$$

where V_n^B is the $\sqsubseteq$-minimum element $V'_n \in \mathbb{S}_n$ with $V'_n \sqsupseteq V_n$ and $V'_n(B) = 1$. The *standard model* is $\mathbb{S} = \{\bigcup \vec{V} : \vec{V} \in \times_{n<\omega} \mathbb{S}_n$ is a $\subseteq -chain\} \cup \{V_\curlywedge\}$.

Fact 7.7. *The standard model* $\mathbb{S}$ *is closed under updates and meets, and so is a* U*-model. Moreover,* $\mathbb{S}$ *is the set of all valuations. Hence,* $\models_U = \models_{\mathbb{S}}$*.*

Proof. Observe first that $\mathbb{S}_n$ is a set of functions $V_n : \mathcal{L}_{U\leq n} \to \{1, 0, \bot\}$. Next we show by induction on n that each V_n satisfies the truth-tables and the update condition $V_n([\pm A]B) = V_n^{[\pm]A}(B)$ over $\mathcal{L}_{U\leq n}$, together with the auxiliary claim that $\mathbb{S}_n$ is closed under meets and updates.

(Base case $n = 0$.) Since $\mathbb{S}_0 = \mathbb{V}$ and $\mathcal{L}_{U0} = \mathcal{L}$, $\mathbb{S}_0$ is a set of $\mathcal{L}_{U0}$-valuations, and it is closed under updates (Proposition 5.2) and meets (Proposition 3.7).

Assume as inductive hypothesis that the claims hold for $\mathbb{S}_n$.

(Ind. case $n+1$) ($\mathbb{S}_{n+1}$ is closed under meets.) Let $\mathbb{S}'_{n+1} \subseteq \mathbb{S}_{n+1}$. We show that $\bigsqcap \mathbb{S}'_{n+1}$ can be built as in Definition 7.6 and so $\bigsqcap \mathbb{S}'_{n+1} \in \mathbb{S}_{n+1}$. To see this, let in Definition 7.6 $V_n = \bigsqcap \mathbb{S}'_n = \bigsqcap\{V_{n+1} \cap (\mathcal{L}_{U\leq n} \times \{1, 0, \bot\}) : V_{n+1} \in \mathbb{S}'_{n+1}\}$. By the inductive hypothesis, V_n and all updates $V_n^B, V_n^{\neg B}$ are in $\mathbb{S}_n$. Then,

$$\begin{aligned} \textstyle\bigsqcap \mathbb{S}'_{n+1}([\pm B]C) &= \textstyle\bigsqcap_{V_{n+1} \in \mathbb{S}'_{n+1}} V_{n+1}([\pm B]C) = \bigsqcap_{V \in \mathbb{S}'_n} (V^B \sqcap V^{\neg B})(C) \\ &= \left(\textstyle\bigsqcap_{V \in \mathbb{S}'_n} V^B \sqcap \bigsqcap_{V \in \mathbb{S}'_n} V^{\neg B}\right)(C) = (V_n^B \sqcap V_n^{\neg B})(C). \end{aligned}$$

For boolean $\mathcal{L}_{Un+1}$-formulas, $\bigsqcap \mathbb{S}'_{n+1}(B * C) \in f_*(\bigsqcap \mathbb{S}'_{n+1}(B), \bigsqcap \mathbb{S}'_{n+1}(C))$ is proved analogously to the propositional case (Proposition 3.4), and similarly for $\neg B$ formulas. The function obtained this way is exactly $\bigsqcap \mathbb{S}'_{n+1}$ and so $\bigsqcap \mathbb{S}'_{n+1} \in \mathbb{S}_{n+1}$. ($\mathbb{S}_{n+1}$ is closed under updates.) For $V_{n+1} \in \mathbb{S}_{n+1}$ and $A \in \mathcal{L}$, let $V_{n+1}^A = \bigsqcap\{V' \in \mathbb{S}_{n+1} : V' \sqsupseteq V_{n+1}$ and $V'(A) = 1\}$. By the claim shown above, V_{n+1}^A is in $\mathbb{S}_{n+1}$ and by definition it is clearly the $\sqsubseteq$-minimum A-refinement of V_{n+1} in $\mathbb{S}_{n+1}$. ($V_{n+1} \in \mathbb{S}_{n+1}$ satisfies the truth-tables.) Since V_{n+1} extends some $V_n \in \mathbb{S}_n$, it respects the truth-tables over all $\mathcal{L}_{U\leq n}$-formulas (by the inductive hypothesis) and also over $\mathcal{L}_{Un+1}$-formulas (by Definition 7.6). This concludes the inductive proof.

Let now $V = \bigcup \vec{V}$ for some $\subseteq$-chain $\vec{V} = \langle V_n \rangle_{n\in\omega}$ with $V_n \in \mathbb{S}_n$. Since each finite fragment V_n satisfies the truth-tables for $\mathcal{L}_{U\leq n}$-formulas, so does

$$\dfrac{\begin{array}{cc}\{A\} & \{\neg A\}\\ \vdots & \vdots\\ B & B\end{array}}{[\pm A]B}\,I[\pm] \qquad \begin{array}{c}\dfrac{A \quad [\pm A]B}{B}\,E[\pm]\\[2ex] \dfrac{\neg A \quad [\pm A]B}{B}\,E[\pm]\end{array} \qquad \begin{array}{c}\dfrac{\neg[\pm A]B}{[\pm A]\neg B}\,\mathit{Red}\neg\\[2ex] \dfrac{[\pm A]\neg B}{\neg[\pm A]B}\,\mathit{Red}'\neg\end{array}$$

Figure 8. An ND system for $\models_U$: consists of the intelim rules (Figure 4) over $\mathcal{L}_U$ plus: an introduction rule $I[\pm]$ (replacing RB); elimination rules $E[\pm]$; and reduction rules for negation $\{\mathit{Red}\neg, \mathit{Red}'\neg\}$. The latter reflect that $[\pm A]$ modalities are functional (Fact 7.9)

V for $\mathcal{L}_U$-formulas. The function V also satisfies the update condition: say $C \in \mathcal{L}_{Un}$; then $V([\pm B]C) = V_{n+1}([\pm B]C) = (V_n^B \sqcap V_n^{\neg B})(C)$. This shows that $\mathbb{S}$ is a set of valuations. ($\mathbb{S}$ is the set of all valuations.) Vice versa, for any valuation $V \neq V_\curlywedge$, V satisfies the truth-tables and the update condition over $\mathcal{L}_U$ and so over every finite language $\mathcal{L}_{Un}$; hence each finite fragment $V_n = V \cap (\mathcal{L}_{U\leq n} \times \{1, 0, \bot\})$ can be built as in Definition 7.6, and so V is of the form $V = \bigcup_n V_n$ for a sequence $\langle V_n \rangle_{n<\omega}$ with $V_n \in \mathbb{S}_n$. ■

Corollary 7.8. (Complete lattice) $(\mathbb{S}, \sqsubseteq^+)$ *is a complete lattice, where* $\sqsubseteq^+ = \sqsubseteq \cup (\mathbb{S} \times \{V_\curlywedge\})$. *Hence, for any* $\mathbb{S}' \subseteq \mathbb{S}$ *its join is* $\bigsqcup \mathbb{S}' = \bigsqcap (\mathbb{S}')^u$.

Fact 7.9. (Functionality) *For any formula* $[\pm A]B \in \mathcal{L}_U$ *and valuation* V, $V([\pm A]B) = V(\langle \pm A \rangle B)$ *for the dual modality* $\langle \pm A \rangle = \neg[\pm A]\neg$.

Proof. The case $V = V_\curlywedge$ is trivial. For any other valuation: $V(\neg[\pm A]\neg B) = 1 \Leftrightarrow V([\pm A]\neg B) = 0 \Leftrightarrow (V^A \sqcap V^{\neg A})(\neg B) = 0 \Leftrightarrow V^A(\neg B) = 0 = V^{\neg A}(\neg B)$ $\Leftrightarrow V^A(B) = 1 = V^{\neg A}(B) \Leftrightarrow (V^A \sqcap V^{\neg A})(B) = 1 \Leftrightarrow V^{[\pm]A}(B) = 1 \Leftrightarrow$ $V([\pm A]B) = 1$. Switching 0s and 1s in this reasoning gives $V(\neg[\pm A]\neg B) = 0$ iff $V([\pm A]B) = 0$. These two cases jointly imply $V(\neg[\pm A]\neg B) = \bot$ iff $V([\pm A]B) = \bot$. In either case, $V(\neg[\pm A]\neg B) = V([\pm A]B)$. ■

Definition 7.10. ($\vdash_U$ *system*) A natural deduction system for $\models_U$ consists of the rules from Figure 8 and the intelim rules (Figure 4) over the language $\mathcal{L}_U$.

Lemma 7.11. *(Soundness of* $\vdash_U$*) The ND system* $\vdash_U$ *is sound with respect to* $\models_U$ *consequence.*

Proof. For an arbitrary U-model $\mathbb{U}$, the soundness of the intelim rules over $\mathcal{L}_U$-formulas follows from its satisfaction of the truth-tables, just as in $\mathcal{L}$.

($I[\pm]$.) Assume that $\Gamma \cup \{A\} \models_{\mathbb{U}} B$ and $\Gamma \cup \{\neg A\} \models_{\mathbb{U}} B$, and let $V \in \mathbb{U}_\Gamma$ be arbitrary. By Definition 7.2, $V^A \in \mathbb{U}$ is the $\sqsubseteq$-minimum element with (i) $V^A \sqsupseteq V$ and (ii) $V^A(A) = 1$. From (i), $V^A[\Gamma] = \{1\}$, and so with (ii)

$$\begin{array}{l} [\pm A]B \\ \hline \mid B \\ \vdots \quad \vdots \\ \mid C \\ \hline [\pm A]C \end{array} \begin{array}{l} O\pm(A) \\ \\ \\ \\ C\pm(A) \end{array} \qquad \frac{[\pm A]B}{\langle\pm A\rangle B}\,D \qquad \frac{\langle\pm A\rangle B}{[\pm A]B}\,\mathit{Alt}$$

$$\frac{[\pm A][\pm B]C}{[\pm B][\pm A]C}\,\mathit{Perm.} \qquad \frac{B}{[\pm A]B}\,\mathit{Triv}$$

$$\frac{[\pm D]B \quad C}{[\pm D]A}\,[\pm]I/E* \qquad \frac{B \quad [\pm D]C}{[\pm D]A}\,[\pm]I/E* \qquad \frac{[\pm A][\pm A]B}{[\pm A]B}\,\mathit{Contr.}$$

$$\frac{([\pm A]^*\langle\pm A\rangle^*)^*B}{([\pm A]^*\langle\pm A\rangle^*)^*B}\,\mathit{Coll.} \qquad \frac{[\pm A]C}{[\pm B]C}\,RE \quad \mathit{whenever}\, A \equiv_0 (\neg)B$$

Figure 9. A list of rules derivable in $\vdash_U$. $O\pm(A), C\pm(A)$ open and close a subproof; any formula above the subproof can be imported into it. $D, \mathit{Alt}, \mathit{Triv}$ are named after modal axioms. *Perm.* is a permutation rule. $[\pm]I*$ and $[\pm]E*$ are modal versions of intelim rules $I*$ or $E*$ $\{B, C\} \therefore A$ with modal premises: $[\pm D]B$ or $[\pm D]C$ or both. *Contraction* and *Collapse* show that any two non-empty sequences of modalities $[\pm A], \langle\pm A\rangle$, indistinctly written $([\pm A]^*\langle\pm A\rangle^*)^*$, are equivalent. (RE) is a rule for the replacement of modalities that are 0-depth equivalent $A \equiv_0 B$, or equivalent to their negations $A \equiv_0 \neg B$

we obtain $V^A[\Gamma \cup \{A\}] = \{1\}$. This implies that $V^A \in \mathbb{U}_{\Gamma\cup\{A\}}$ and using the initial assumption $V^A(B) = 1$. With a similar reasoning for $V^{\neg A}$, we conclude $V^A(B) \sqcap V^{\neg A}(B) = 1$ and so $V([\pm A]B) = 1$.

$(E[\pm].)$ Let $V \in \mathbb{U}$ be such that $V([\pm A]B) = 1$ and $V(A) = 1$. As in the propositional case (Fact 5.6) the definition of update gives $V^A = V$ and $V^{\neg A} = V_\curlywedge$ and so $V^{[\pm]A} = V$. From this, we obtain $1 = V([\pm A]B) = V^{[\pm]A}(B) = V(B)$. The proof for the $\neg A$ case is analogous.

$(\mathit{Red}\neg, \mathit{Red}'\neg.)$ We use Fact 7.9: $V([\pm A]\neg B) = 1$ iff $V(\neg[\pm A]\neg B) = 0$ iff $V([\pm A]B) = 0$ iff $V(\neg[\pm A]B) = 1$. ■

See Figure 9 for a list of additional rules, also written $\{B, C\} \therefore A$ in text. Permutation for $\mathcal{L}$-valuations (Corollary 5.9) extends to $\mathcal{L}_U$, showing that *Perm.* is sound, and the soundness of D and *Alt* follows from Fact 7.9. In fact, all the rules in Figure 9 are derivable in $\vdash_U$. The RE rule replaces update modalities $[\pm A]$ and $[\pm B]$ if A is 0-depth equivalent to B (written $A \equiv_0 B$) or to its negation $\neg B$, where $A \equiv_0 B$ iff $A \vdash_0 B$ and $B \vdash_0 A$.

Example 10. (*Modal intelim rules*) Let us compare the intelim rule $E\vee$ over $\mathcal{L}_U$ with its modal counterpart, the rule $[\pm]E\vee$:

$$\frac{([\pm D]A \vee B) \quad \neg B}{\neg[\pm D]A}E\vee \qquad\qquad \frac{[\pm D](A \vee B) \quad \neg B}{[\pm D]\neg A}\,[\pm]E\vee$$

$E\vee$ treats the modal formula $[\pm D]A$ as a block, while $[\pm]E\vee$ ignores the prefix $[\pm D]$, applies $E\vee$, and then puts $[\pm D]$ back into the conclusion.

PROPOSITION 7.12. *(Derived rules) The rules in Figure 9 are derivable in* $\vdash_U$.

PROOF. $O\pm(A), C\pm(A)$. Let Π be a proof of C from $\{B\} \cup \Delta$. Prefix Π with an application of $E[\pm]$ with A to obtain a proof Π_1 of C from $\{A, [\pm A]B\} \cup \Delta$. A similar reasoning with $\neg A$ gives also a proof Π_2 of C from $\{\neg A, [\pm A]B\} \cup \Delta$. Combine Π_1, Π_2 and an instance of $I[\pm]$ over $\{A, \neg A\}$ for a proof of $[\pm A]C$ from $\{[\pm A]B\} \cup \Delta$. Note that Δ is imported into the subproof.
(D). From $[\pm A]B$, an application of $O\pm(A)$, $C\pm(A)$ over $B \vdash_U \neg\neg B$ gives a proof of $[\pm A]\neg\neg B$. Now, applying $Red'\neg$ we obtain $\neg[\pm A]\neg B$, i.e. $\langle\pm A\rangle B$.
(Alt). From $\neg[\pm A]\neg B$, an application of $Red\neg$ gives $[\pm A]\neg\neg B$, and an application of $O\pm(A), C\pm(A)$ over $\neg\neg B \vdash_U B$ gives us a proof of $[\pm A]B$.
$(Perm.)$. From $[\pm A][\pm B]C$, apply $E[\pm]$ with A and then again with B, to obtain a proof $\Pi_{A,B}$ of C from $\{[\pm A][\pm B]C\} \cup \{A, B\}$. Similar proofs $\Pi_{A \neg B}$, $\Pi_{\neg A,B}$ and $\Pi_{\neg A,\neg B}$ of C are obtained from the corresponding assumptions. Applying $I[\pm]$ to $\Pi_{A,B}, \Pi_{\neg A,B}$ and in parallel to $\Pi_{A,\neg B}, \Pi_{\neg A,\neg B}$ gives two proofs of $[\pm A]C$ from $\{[\pm A][\pm B]C\} \cup \{B\}$ resp. $\{[\pm A][\pm B]C\} \cup \{\neg B\}$ discharging $\{A, \neg A\}$; a final application of $I[\pm]$ to the latter proofs gives us a proof of $[\pm B][\pm A]C$ from $\{[\pm A][\pm B]C\}$ discharging $\{B, \neg B\}$.
$(Triv)$. A proof of B from Γ is also a proof of B from $\Gamma \cup \{A\}$ and from $\Gamma \cup \{\neg A\}$. An application of $I[\pm]$ to the latter proofs that vacuously discharges $\{A, \neg A\}$ gives a proof of $[\pm A]B$ from Γ.
$([\pm]I*)$ or $([\pm]E*)$. The derivability of each modal intelim rule can be routinely checked. We only illustrate the case $[\pm]E\vee$ above. From $\{[\pm D](A \vee B), \neg B\}$, open a subproof with $A \vee B$, import $\neg B$ and apply $E\vee$ to conclude A. Closing this subproof gives us $[\pm D]A$. (If both premises are in the scope of $[\pm D]$, we obtain $[\pm D][\pm D]A$ and apply *Contr.*, see below, to get $[\pm D]A$.)
$(Contr.)$. Applying $E[\pm]$ twice with A to the premise gives a proof of B from $\{A, [\pm A][\pm A]B\}$. Similarly, we get a proof of B from $\{\neg A, [\pm A][\pm A]B\}$. An application of $I[\pm]$ now gives a proof of $[\pm A]B$ from $\{[\pm A][\pm A]B\}$.
$(Coll.)$. Any pair of formulas of the form $([\pm A]^*\langle\pm A\rangle^*)^*B$ can be transformed into each other by repeated applications of the following rules (in the worst case leading to $[\pm A]B$ as an intermediate step):

$$\begin{tikzcd}
{[\pm A]} \arrow[r, bend left, "Triv"] \arrow[d, bend left, "D"] & {[\pm A][\pm A]} \arrow[l, bend left, "Contr"] \arrow[r, bend left, "D"] & {\langle\pm A\rangle[\pm A]} \arrow[l, bend left, "Alt"] \\
{\langle\pm A\rangle} \arrow[u, bend left, "Alt"] \arrow[r, "Triv"] & {[\pm A]\langle\pm A\rangle} \arrow[u, bend right, "(\star)"] \arrow[r, bend left, "D"] & {\langle\pm A\rangle\langle\pm A\rangle} \arrow[l, bend left, "Alt"]
\end{tikzcd}$$

where $(\star)$ turns $[\pm A]\langle \pm A\rangle B$ into two proofs of $\langle \pm A\rangle B$ from $\Gamma \cup \{A\}$, resp. $\Gamma \cup \{\neg A\}$ (using $E[\pm]$), followed by *Alt* in each proof, thus concluding $[\pm A]B$; a final application of $I[\pm]$ gives us a proof of $[\pm A][\pm A]B$ from Γ.
(*RE*). First note that $A \equiv_0 B$ implies $B \vdash_0 A$ and $\neg B \vdash_0 \neg A$. To see this, recall that $\vdash_0$ has no theorems so a (non-redundant) proof Π_{AB} of $A \vdash_0 B$ exists consisting solely of single premise rules. Since all these rules $C \mathrel{.\!\raisebox{1ex}{.}\!.} D$ in Figure 4 admit contraposition $\neg D \mathrel{.\!\raisebox{1ex}{.}\!.} \neg C$, reversing this proof gives a proof $\Pi_{\neg B \neg A}$ of $\neg B \vdash_0 \neg A$. Let then Π be a proof of $[\pm A]C$. Combined with Π_{BA}, an application of $E[\pm]$ gives a proof Π_1 of C from B. Another combination with $\Pi_{\neg B \neg A}$ gives a proof Π_2 of C from $\neg B$. Finally, combine Π_1, Π_2 and an application of $I[\pm]$ for a proof of $[\pm B]C$ discharging $\{B, \neg B\}$.

The case where $A \equiv_0 \neg B$ is similar: combine Π with each proof $\Pi_{\neg BA}$ and $\Pi_{\neg\neg B \neg A}$ and finally $I[\pm]$ to obtain $[\pm \neg B]C$.[13] ■

It can also be pointed out that $\vdash_U$ enjoys a modal deduction theorem $\Gamma \vdash_U [\pm A]B$ iff $\Gamma \cup \{\pm A\} \vdash_U B$. Basic modal axioms are not valid, though[14]:

$$\begin{array}{rl} \textit{Axiom } \mathsf{K} \text{ fails:} & \not\models_U [\pm p](p \to p) \to ([\pm p]p \to [\pm p]p)^{14} \\ \textit{Distribution} \text{ fails:} & \not\models_U \langle \pm p\rangle(p \vee \neg p) \to (\langle \pm p\rangle p \vee \langle \pm\rangle \neg p). \end{array}$$

8. Completeness and Conservativeness of the $\vdash_U$ Logic

We define next the closure of a set Γ under all $\vdash_U$-rules except for $I[\pm]$. This rule is added at the construction of minimum functions V_Γ in Definition 8.2.

Definition 8.1. (*Weak closure*; $\perp$-*completion*) Define the *weak closure* of a set $\Gamma \subseteq \mathcal{L}_U$, denoted $\mathsf{Cn}_{U0}(\Gamma)$, as the $\subseteq$-minimal set such that: $\Gamma \subseteq \mathsf{Cn}_{U0}(\Gamma)$ and $\mathsf{Cn}_{U0}(\Gamma)$ is closed under any instance of $\{I*, E*, E[\pm], \mathit{Red}\neg, \mathit{Red}'\neg\}$.

Define the $\perp$-*completion* of a partial function $f : \mathcal{L}_U \to \{1, 0, \perp\}$ as $\perp(f) = f \cup \{\langle A, \perp\rangle : A \notin \mathrm{dom}(f)\}$.

Definition 8.2. ($\vdash_U$-*closed function*) Define simultaneously for all sets $\Delta \subseteq \mathcal{L}_U$ each level V^n_Δ of the $\vdash_U$-*closed function* of Δ by:

$$\begin{aligned} V^0_\Delta &= (\mathsf{Cn}_{U0}(\Delta) \times \{1\}) \cup \{\langle B, 0\rangle : \neg B \in \mathsf{Cn}_{U0}(\Delta)\} \\ V^{n+1}_\Delta &= (\mathsf{Cn}_{U0}(I[\pm](\Delta, n)) \times \{1\}) \cup (\{B : \neg B \in \mathsf{Cn}_{U0}(I[\pm](\Delta, n))\} \times \{0\}) \end{aligned}$$

[13] In the present setting, the *RE* rule is covered by these two cases $A \equiv_0 B$ and $A \equiv_0 \neg B$. More instances of *RE* hold if e.g. *idempotence* rules $A \vee A \mathrel{.\!\raisebox{1ex}{.}\!.} A$ are added to $\vdash_0$.

[14] A version of axiom K as a rule $\{[\pm A](B \to C), [\pm A]B\} \mathrel{.\!\raisebox{1ex}{.}\!.} [\pm A]C$ is valid, though, as a modal intelim rule $[\pm]E{\to}$.

where $I[\pm](\Delta, n) = \mathsf{Th}(V^n_\Delta) \cup \bigcup_{A \in \mathcal{L}} \{[\pm A]B : \langle B, 1\rangle \in V^n_{\Delta\cup\{A\}} \cap V^n_{\Delta\cup\{\neg A\}}\}$. Finally, define: $V^\omega_\Delta = \bigcup_n V^n_\Delta$ and $V_\Delta = \bot(V^\omega_\Delta)$.

Definition 8.3. (*Canonical model* $\mathbb{M}$) The *canonical model* is the set of all $\vdash_U$-closed functions $\mathbb{M} = \{V_\Gamma : \Gamma \subseteq \mathcal{L}_U\}$ built using Definition 8.2. Its $\sqsubseteq$-minimum element is $V_\emptyset$. Its $\sqsubseteq$-maximum element is $V_{\{p\wedge\neg p\}}$, abusively denoted $V_\curlywedge$.

Lemma 8.4. (Truth lemma) *For any* $\Gamma \cup \{A\} \subseteq \mathcal{L}_U$*,* $\Gamma \vdash_U A$ *iff* $V_\Gamma(A) = 1$.

Proof. ($\Rightarrow$). Let Π be a proof of A from Γ, and let $\Pi_1, \ldots, \Pi_m$ be all the subtrees of Π with an instance of $I[\pm]$ at the root node, say for a conclusion $[\pm B_i]C_i$ in Π_i. Clearly, for each $1 \leq i, j \leq m$ either $\Pi_i \not\subseteq \Pi_j$ or $\Pi_i \not\subseteq \Pi_j$ or both, so assume an ordering $\Pi_1, \ldots, \Pi_m$ satisfying: all subtrees of Π_{i+1} also applying $I[\pm]$ are among $\{\Pi_1, \ldots, \Pi_i\}$. Let also $\langle A_1, \ldots, A_n\rangle$ be a sequence with all formula labels (possibly repeated) in Π that respects the inverse tree order: if $\{B, C\} \therefore A$ occurs in Π, then B and C are listed before A in the sequence. Let us rephrase this sequence $\langle A_1, \ldots, A_n\rangle$ to mark each conclusion $[B_i]C_i$ of $I[\pm]$ as a node A^i_0:

$$\langle A^0_1, \ldots, A^0_{n_1}, A^1_0 = [B_1]C_1, A^1_1, \ldots, A^1_{n_1}, \ldots, A^m_0 = [B_m]C_m, A^m_1, \ldots, A^m_{n_m}\rangle$$

From the original proof Π for $\Gamma \vdash_U A$, one easily obtains a proof of A from any set of the form $\Gamma \cup \{\pm B_1, \ldots, \pm B_m\}$. We show by induction that for each A^i_j in the above sequence, $V^i_{\Gamma\cup\{\pm B_{i+1},\ldots,\pm B_m\}}(A^i_j) = 1$ (where V^i is as in Definition 8.2). The proof will conclude with the goal $V^m_{\Gamma\cup\emptyset}(A) = V^m_\Gamma(A) = 1$ since $A = A^m_{n_m}$. For convienience let $\Gamma(i)$ denote any extension of Γ with choices among $\{\pm B_i, \ldots, \pm B_m\}$; that is, $\Gamma(i) = \Gamma \cup \{\pm B_i, \ldots, \pm B_m\}$.
(Base case A_1.) In case $A_1 = A^0_1 \in \Gamma(1)$ we simply have: $V^0_{\Gamma(1)}(A_1) = 1$. Otherwise, $A_1 = A^1_0 = [\pm B_1]C_1$ obtains by $I[\pm]$ in which case $\Gamma(2) \cup \{B_1\} \vdash_U C_1$ and $\Gamma(2) \cup \{\neg B_1\} \vdash_U C_1$. Since Π_1 contains no other instance of $I[\pm]$, there are $\vdash_U$-proofs Π'_1, Π''_1 of these two claims without $I[\pm]$. Thus, $V^0_{\Gamma(2)\cup\{B_1\}}(C_1) = 1 = V^0_{\Gamma(2)\cup\{\neg B_1\}}(C_1)$ and therefore $V^1_{\Gamma(2)}([\pm B_1]C_1) = 1$.
(Ind. case A_{k+1}.) Assume as inductive hypothesis that for each $A_{k'} = A^i_j$ with $k' \leq k$ it holds that $V^i_{\Gamma(i+1)}(A^i_j) = 1$. Let now $A_k = A^i_j$. In case $A_{k+1} = A^i_{j+1}$, then A_{k+1} follows in Π from an intelim rule, $E[\pm]$, *Red*$\neg$ or *Red*$'\neg$ rule from $\{A_1, \ldots, A_k\}$, in which case $V^i_{\Gamma(i)}(A_{k+1}) = 1$. Otherwise, $A_{k+1} = A^{i+1}_0 = [\pm B_{i+1}]C_{i+1}$ and this formula is derived in Π using $I[\pm]$ from any set of the form $\Gamma(i+2) = \Gamma \cup \{\pm B_{i+2}, \ldots, \pm B_m\}$. Thus there are proofs of C_{i+1} from $\Gamma(i+2) \cup \{B_{i+1}\}$ and from $\Gamma(i+2) \cup \{\neg B_{i+1}\}$. By the construction of the sequence, any such instance C_{i+1} occurs before $[\pm B_{i+1}]C_{i+1} = A^{i+1}_0$, and so in the worst case it is of the form

$C_{i+1} = A^i_j$. By the inductive hypothesis $V^i_{\Gamma(i+1)}(C_{i+1}) = 1$. In particular, $V^i_{\Gamma(i+2)\cup\{B_{i+1}\}}(C_{i+1}) = 1$ and $V^i_{\Gamma(i+2)\cup\{\neg B_{i+1}\}}(C_{i+1}) = 1$. Thus, $\langle C_{i+1}, 1\rangle \in V^i_{\Gamma(i+2)\cup\{B_{i+1}\}} \cap V^i_{\Gamma(i+2)\cup\{\neg B_{i+1}\}}$ and so by definition $[\pm B_{i+1}]C_{i+1} \in I[\pm](\Gamma(i+2), i)$ and finally $V^{i+1}_{\Gamma(i+2)}([\pm B_{i+1}]C_{i+1}) = 1$.
($\Leftarrow$). This is immediate, since V_Γ is built by assigning 1 to every formula in Γ and closing the set of 1-values under $\vdash_U$-rules, following Definitions 8.1–8.2.

■

Lemma 8.5. (Truth-tables and $\vdash_U$-closure) *For any function* $V_\Gamma \in \mathbb{M}$ *with* $V_\Gamma \neq V_\curlywedge$, V_Γ *satisfies the truth-tables over* $\mathcal{L}_U$.

Proof. ($\neg$). First, $V_\Gamma(\neg A) = 1$ implies $V_\Gamma(A) = 0$ by Definition 8.2; secondly, an assignment $V_\Gamma(\neg A) = 0$ occurs only if $V_\Gamma(\neg\neg A) = 1$, in which case $V_\Gamma(A) = 1$ by closure under $E\neg$; finally, the case $\langle A, \bot\rangle \in V_\Gamma$ implies that $\langle A, 1\rangle, \langle A, 0\rangle \notin V^\omega_\Gamma$ and by the above reasoning $\langle\neg A, 0\rangle, \langle\neg A, 1\rangle \notin V^\omega_\Gamma$. Hence $\langle\neg A, \bot\rangle \in \bot(V^\omega_\Gamma) = V_\Gamma$. In either case, $V_\Gamma(\neg A) \in f_\neg(V_\Gamma(A))$.
($* \in \{\wedge, \vee, \to\}$). By induction on the complexity of formulas $B, C \in \mathcal{L}_U$.

(Base case $B, C \in \mathcal{L}$.) Let $B, C \in \mathcal{L} = \mathcal{L}_{U0}$. Since V_Γ is closed under intelim rules, $V^{\mathcal{L}}_\Gamma = V_\Gamma \cap (\mathcal{L} \times \{1, 0, \bot\})$ is a function over $\mathcal{L}$ closed under intelim rules over $\mathcal{L}$. It is also $\vdash_0$-consistent, since V_Γ is $\vdash_U$-consistent and $\vdash_0 \subseteq \vdash_U$. By Proposition 3.8, $V^{\mathcal{L}}_\Gamma$ is a valuation over $\mathcal{L}$ and so satisfies the truth-tables over $\mathcal{L}$. Since $V^{\mathcal{L}}_\Gamma \subseteq V_\Gamma$, V_Γ also satisfies the truth-tables over $\mathcal{L}$.

(Ind. case $B, C \in \mathcal{L}_{U\leq k+1}$.) Suppose as inductive hypothesis that V_Γ satisfies the truth-tables over $\mathcal{L}_{U\leq k}$-formulas. Following Definition 7.5, let $B = [\pm A_1]B'$ and $C = [\pm A_2]C'$ be basic $\mathcal{L}_{Uk+1}$-formulas (if B or C is in $\mathcal{L}_{U\leq k}$ the proof is analogous). We only show the case $* = \vee$, the other cases can be proved with similar arguments.
(Case $B \mapsto 1$ or $C \mapsto 1$.) V_Γ is closed under $I\vee$, so $V_\Gamma(B \vee C) = 1$.
(Case $B \mapsto 0$ or $C \mapsto 0$.) We only show the former and prove $V_\Gamma(B \vee C) = V_\Gamma(C)$, so let $V_\Gamma(B) = V_\Gamma([\pm A_1]B') = 0$. By the previous proof on negation, $V_\Gamma(\neg B) = V_\Gamma(\neg[\pm A_1]B') = 1$. Consider the value of $B \vee C$:

(Subcase $B \vee C \mapsto 1$.) By *Triv*, $V_\Gamma([\pm A_2]B \vee C) = 1$, so we can apply:

$$\frac{\neg[\pm A_1]B' \quad [\pm A_2]([\pm A_1]B' \vee [\pm A_2]'C')}{[\pm A_2][\pm A_2]C'}\,[\pm]E\vee$$

A further application of *Collapse* gives $V_\Gamma([\pm A_2]C') = V_\Gamma(C) = 1$.

(Subcase $B \vee C \mapsto 0$.) Thus, $V_\Gamma(\neg(B \vee C)) = 1$ and by the intelim rule $E\vee$ we obtain $V_\Gamma(\neg C) = 1$, so $V_\Gamma(C) = 0$.

(Subcase $B \vee C \mapsto \bot$.) One can see that $V_\Gamma(C) \neq 1$ as otherwise $V_\Gamma(B \vee C) = 1$. Moreover, $V_\Gamma(C) \neq 0$, as otherwise $V_\Gamma(\neg C) = 1$ and with the case assumption $V_\Gamma(\neg B) = 1$ and $I\vee$ would give $V_\Gamma(\neg(B \vee C)) = 1$, contradicting the subcase assumption. Thus, it must be that $V_\Gamma(C) = \bot$.

In all subcases, $V_\Gamma(B \vee C) = V_\Gamma(C)$, and so V_Γ satisfies the truth-table $f_\vee(0, v) = v$ for the case $B \mapsto 0$.
(Case $B \mapsto \bot$ and $C \mapsto \bot$.) We only need to show that $V_\Gamma(B \vee C) \neq 0$, but this is immediate since otherwise $V_\Gamma(\neg(B \vee C)) = 1$, and then we would have $V_\Gamma(\neg B) = 1$ and finally $V_\Gamma(B) = 0$, contradicting the case assumption.

That $V_\Gamma(B \vee C) \in f_\vee(B, C)$ holds for non-basic formulas $B \vee C \in \mathcal{L}_{U \leq k+1}$ is proved as in Proposition [11, 2.26] but with intelim rules over $\mathcal{L}_U$. This inductive proof shows that V_Γ respects the truth-tables for any $\mathcal{L}_U$-formula. ■

Lemma 8.6. (Canonicity) *The canonical model* $\mathbb{M}$ *is a* U*-model. The* A*-update of a valuation* V_Γ *is the valuation* $V_\Gamma^A = V_{\Gamma \cup \{A\}}$.

Proof. Any $V_\Gamma \neq V_\curlywedge$ satisfies the truth-tables as shown in Lemma 8.5. To see that $\mathbb{M}$ is also closed under updates, let $V_\Gamma^A = V_{\Gamma \cup \{A\}}$. This satisfies (i) $V_\Gamma^A \sqsupseteq V_\Gamma$ and (ii) $V_\Gamma^A(A) = 1$. For any other V_Δ satisfying the corresponding (i)–(ii), it must be that $V_\Delta[\Gamma \cup \{A\}] = 1$ and so by $\vdash_U$-closure, $V_{\Gamma \cup \{A\}} \sqsubseteq V_\Delta$. Thus, V_Γ^A is minimal with (i)–(ii). Since $V_{\Gamma \cup \{A\}} \in \mathbb{M}$, we are done. ■

Theorem 8.7. (Completeness) *For* $\Gamma \cup \{A\} \subseteq \mathcal{L}_U$*,* $\Gamma \models_U A$ *implies* $\Gamma \vdash_U A$.

Proof. Let us prove the contrapositive. Assume $\Gamma \nvdash_U A$. By Lemma 8.4, $V_\Gamma(A) \neq 1$ and so there is $V \in \mathbb{M}$ with $V[\Gamma] = \{1\}$ and $V(A) \neq 1$. By Definition 7.3 and Lemma 8.6, we obtain $\Gamma \not\models_{\mathbb{M}} A$ and finally $\Gamma \not\models_U A$. ■

Lemma 8.8. (Lower bound) *For any set* $\Gamma \subseteq \mathcal{L}_U$ *and valuation* V *it holds that:* $V_\Gamma \sqsubseteq V$ *iff* $V \in \mathbb{S}_\Gamma$.

Proof. ($\Rightarrow$). Assume $V_\Gamma \sqsubseteq V$. Since $V_\Gamma[\Gamma] = \{1\}$, it also holds that $V[\Gamma] = \{1\}$. This and $V \in \mathbb{S}$ (Fact 7.7) imply $V \in \mathbb{S}_\Gamma$. ($\Leftarrow$). Ignoring the trivial case $V = V_\curlywedge$, we show $V_\Gamma(B) \leq V(B)$ for any $B \in \mathcal{L}_U$. For $V_\Gamma(B) = 1$, Lemma 8.4 gives $\Gamma \vdash_U B$ and by soundness, $\Gamma \models_U B$; in particular $\Gamma \models_{\mathbb{S}} B$. Hence, the assumption $V \in \mathbb{S}_\Gamma$ implies $V(B) = 1$. The case $V_\Gamma(B) = 0$ reduces to the former case with the formula $\neg B$. The case $V_\Gamma(B) = \bot$ is immediate. ■

Some of the previous results for the propositional case extend from $\mathcal{L}$ to $\mathcal{L}_U$ with minimal changes on proofs. This is the case of:

- (from Lemma 4.3) for any $V, V' \in \mathbb{S}$, $\mathsf{Th}(V) \subseteq \mathsf{Th}(V')$ iff $V \sqsubseteq V'$;
- (from Corollary 4.10) $V_\Gamma = V_{\mathsf{Cn}_U(\Gamma)}$, where $\mathsf{Cn}_U(\Gamma) = \{A \in \mathcal{L}_U : \Gamma \vdash_U A\}$.

Corollary 8.9. (Correspondence) *(i) For any valuation* $V' = V_{\mathsf{Th}(V')}$*; (ii) for any* $\Gamma \subseteq \mathcal{L}_U$*,* $V_\Gamma = \bigsqcap \mathbb{S}_\Gamma$.

Proof. For (i), $V_{\mathsf{Th}(V')}(A) = 1$ iff $\mathsf{Th}(V') \vdash_U A$ (by Lemma 8.4) iff $V'(A) = 1$ where in the latter equivalence, ($\Rightarrow$) follows from V' being closed under $\vdash_U$-rules and ($\Leftarrow$) is immediate from def. of $\mathsf{Th}(V')$.

For (ii). ($\sqsubseteq$) From $V_\Gamma \sqsubseteq V$ iff $V \in \mathbb{S}_\Gamma$ (Lemma 8.8), it is immediate that: $V_\Gamma \sqsubseteq \bigsqcap \mathbb{S}_\Gamma$. ($\sqsupseteq$). Assume $(\bigsqcap \mathbb{S}_\Gamma)(A) = 1$. Since $\mathbb{S}$ contains all valuations (Fact 7.7), for any U-model $\mathbb{U}$, $\mathbb{U}_\Gamma \subseteq \mathbb{S}_\Gamma$ and thus $1 = (\bigsqcap \mathbb{S}_\Gamma)(A) \leq (\bigsqcap \mathbb{U}_\Gamma)(A)$. Hence, $\Gamma \models_{\mathbb{U}} A$ for any U-model $\mathbb{U}$ and so, by definition, $\Gamma \models_U A$; then, by completeness $\Gamma \vdash_U A$, which finally gives $V_\Gamma(A) = 1$. Since Lemma 4.3 also holds for $\mathcal{L}_U$-valuations, we obtain $V_\Gamma \sqsupseteq \bigsqcap \mathbb{S}_\Gamma$. ■

Corollary 8.10. (Canonical is standard) *The canonical model is the standard model:* $\mathbb{M} = \mathbb{S}$. *The* $\sqsubseteq$*-minimum valuation is* $\bigsqcap \mathbb{S} = V_\emptyset$.

Proof. The inclusion $\mathbb{M} \subseteq \mathbb{S}$ follows from $\mathbb{M}$ being a U-model (Lemma 8.6) and all valuations being elements of the standard model $\mathbb{S}$. Corollary 8.9 implies the inclusion $\mathbb{S} \subseteq \mathbb{M}$ since for every valuation V', $V' = V_{\mathsf{Th}(V')}$ and thus $V' \in \mathbb{M}$. $\bigsqcap \mathbb{S} = \bigsqcap \mathbb{S}_\emptyset = V_\emptyset$ is immediate from Corollary 8.9 as well. ■

Corollary 8.11. (Trivial [±]-updates) *If* $V(A) \neq \bot$ *then* $V^{[\pm]A} = V$.

Proof. Let $V' \in \mathbb{S}$. Consider first the case $V'(A) = 1$. Then,

$$\begin{aligned} V'^A &= V^A_{\mathsf{Th}(V')} = V_{\mathsf{Th}(V') \cup \{A\}} = V_{\mathsf{Cn}_U(\mathsf{Th}(V') \cup \{A\})} = \\ &= V_{\mathsf{Cn}_U(\mathsf{Th}(V'))} = V_{\mathsf{Th}(V')} = V'. \end{aligned}$$

where each identities follows resp. from: Corollary 8.9; Lemma 8.6; Corollary 4.10 for $\mathcal{L}_U$; the assumption $A \in \mathsf{Th}(V')$ implying $\mathsf{Th}(V') = \mathsf{Th}(V') \cup \{A\}$; and again Corollary 4.10 and Corollary 8.9. Moreover, $\mathsf{Th}(V') \cup \{\neg A\} \vdash_U A, \neg A$ and so $V^{\neg A}_{\mathsf{Th}(V')} = V_{\mathsf{Th}(V') \cup \{\neg A\}} = V_\curlywedge$. In summary, $V'^{[\pm]A} = V'^A \sqcap V'^{\neg A} = V' \sqcap V_\curlywedge = V'$. For the case $V'(A) = 0$, we similarly have: $V'^{[\pm]A} = V_\curlywedge \sqcap V' = V'$. ■

Definition 8.12. (*Translation*) We define a translation function $t : \mathcal{L}_U \to \mathcal{L}$ that deletes all modalities as follows:

$$\begin{array}{cc} t(p) = p & t(\neg A) = \neg t(A) \\ t(A * B) = t(A) * t(B) & t([\pm A]B) = t(B) \end{array}$$

where $* \in \{\wedge, \vee, \to\}$. Below we also use $t\Gamma$ and tB instead of $t[\Gamma]$ and $t(B)$.

Definition 8.13. (*Two-valued valuations*) Define $\mathbb{C} \subseteq \mathbb{S}$ as the set of valuations $\overline{V} : \mathcal{L}_U \to \{1, 0\}$. For $\Gamma \subseteq \mathcal{L}_U$, define also $\mathbb{C}_\Gamma = \{\overline{V} \in \mathbb{C} : \overline{V}[\Gamma] = 1\}$.

Lemma 8.14. (Classical coincidence) *For any formula* $A \in \mathcal{L}_U$ *and valuation* $\overline{V} : \mathcal{L}_U \to \{1, 0\}$, $\overline{V}(A) = \overline{V}(tA)$. *As a consequence,* $\mathbb{C}_\Gamma = \mathbb{C}_{t\Gamma}$.

Proof. By induction on A. Given that $\overline{V}$ is a two-valued valuation, the truth-tables reduce to the classical case, and so the claim holds for boolean operators $A * B$ and $\neg A$. The modal case $[\pm A]B$ follows from Corollary 8.11. ■

Corollary 8.15. *For any* $\Gamma \cup \{B\} \subseteq \mathcal{L}_U$, $\Gamma \models_U B$ *implies* $t\Gamma \models tB$.

Proof. $\mathbb{S}$-valuations build from propositional ones in $\mathbb{V}$ (Definition 7.6), so one can rephrase classical consequence as: $\Delta \models A$ iff $\overline{V}(A) = 1$ for all $\overline{V} \in \mathbb{C}_\Delta$. Then, we reason as follows:

$$\begin{aligned}
\Gamma \models_U B &\Rightarrow V(B) = 1 \text{ for all } V \in \mathbb{S}_\Gamma && \text{(Fact 7.7)}\\
&\Rightarrow \overline{V}(B) = 1 \text{ for all } \overline{V} \in \mathbb{C}_\Gamma \subseteq \mathbb{S}_\Gamma &&\\
&\Rightarrow \overline{V}(tB) = 1 \text{ for all } \overline{V} \in \mathbb{C}_{t\Gamma} && \text{(Lemma 8.14)}\\
&\Rightarrow t\Gamma \models tB && (t\Gamma \cup \{tB\} \subseteq \mathcal{L}).
\end{aligned}$$

■

Lemma 8.16. (Conservativeness) *For* $\Gamma \cup \{A\} \subseteq \mathcal{L}$, $\Gamma \models_U A$ *implies* $\Gamma \models_0 A$.

Proof. Towards a contradiction, assume $\Gamma \models_U A$ but $\Gamma \not\models_0 A$. Since $t\Gamma = \Gamma$ and $tA = A$, by Corollary 8.15 we know that $\Gamma \models A$, i.e. $\Gamma \vdash A$, so let Π be a proof of this. Without loss of generality, assume Π contains exactly one instance of RB. (Otherwise, isolate a subproof in Π with one instance of RB, say for $\Delta \vdash_1 A'$ and reason similarly.) From $\Gamma \models_U A$ and $\Gamma \not\models_0 A$, there is some $V \in \mathbb{V}_\Gamma$ with $V(A) \neq 1$. As in the proof of Fact 7.7, extend the countermodel $V \in \mathbb{V} = \mathbb{S}_0$ into a valuation V' in $\mathbb{S}$. Obviously, $V'[\Gamma] = 1$ and $V'(A) \neq 1$, so $\Gamma \not\models_{\mathbb{S}} A$, and therefore $\Gamma \not\models_U A$ (contradiction). ■

When modal formulas are considered, though, $\models_U$ validates a proof blueprint $[\pm A_1] \dots [\pm A_k]B$ for every theorem B of classical logic.

Proposition 8.17. (Degrees and updates) *For any* $\Gamma \cup \{B\} \subseteq \mathcal{L}$,

$$\Gamma \Vdash^{\mathsf{f}}_k B \textit{ iff } \Gamma \models_U [\pm A_1] \dots [\pm A_k]B \textit{ for some } A_1, \dots, A_k \in \mathsf{f}(\Gamma \cup \{B\}).$$

Proof. By induction on k. (Base Case 0.) The $\Rightarrow$ direction is immediate and $\Leftarrow$ follows from Lemma 8.16. (Ind. case $k+1$.) Assume as inductive hypothesis that the claim holds for k. We omit everywhere that $A_1, \dots, A_k \in \mathsf{f}(\Gamma \cup \{B\})$ and that V ranges over $\mathbb{S}_\Gamma$ unless specified otherwise:

$$\begin{aligned}
&\Gamma \models_U [\pm A_1] \dots [\pm A_{k+1}]B && \Leftrightarrow V([\pm A_1] \dots [\pm A_{k+1}]B) = 1\\
&\Leftrightarrow V^{[\pm]A_1}([\pm A_2] \dots [\pm A_{k+1}]B) = 1 && \Leftrightarrow V^{\pm A_1}([\pm A_2] \dots [\pm A_{k+1}]B) = 1\\
&\Leftrightarrow V([\pm A_2] \dots [\pm A_{k+1}]B) = 1 && \text{for all } V \in (\mathbb{S}_\Gamma)^{A_1} \cup (\mathbb{S}_\Gamma)^{\neg A_1}\\
&\Leftrightarrow V([\pm A_2] \dots [\pm A_{k+1}]B) = 1 && \text{for all } V \in \mathbb{S}_{\Gamma \cup \{A_1\}} \cup \mathbb{S}_{\Gamma \cup \{\neg A_1\}}\\
&\Leftrightarrow \Gamma \cup \{\pm A_1\} \models_U [\pm A_2] \dots [\pm A_{k+1}]B && \Leftrightarrow \Gamma \cup \{\pm A_1\} \Vdash^{\mathsf{f}}_k B\\
&\Leftrightarrow \Gamma \Vdash^{\mathsf{f}}_{k+1} B.
\end{aligned}$$

where $(\mathbb{S}_\Gamma)^{A_1} = \mathbb{S}_{\Gamma\cup\{A_1\}}$ follows from $V'^{A_1} = V^{A_1}_{\mathsf{Th}(V')} = V_{\mathsf{Th}(V')\cup\{A_1\}}$ and so V' is in one set iff it is in the other. (Similarly for $(\mathbb{S}_\Gamma)^{\neg A_1} = \mathbb{S}_{\Gamma\cup\{\neg A_1\}}$.) ■

9. Complexity of the Modal Logic $\vdash_U$ and Applications

Let us address next the computational complexity of the decision problems $\textsc{Val}_U$ and $\textsc{MC}^{\emptyset}_U$ of *validity* and resp. *model checking* for formulas in the minimum valuation $V_\emptyset$. See [35,41] for further details.

Definition 9.1. (*Decision problems*) The set $\textsc{Val}_U \subseteq \mathcal{L}_U$ is defined by: $A \in \textsc{Val}_U$ iff for all $V \in \mathbb{S}$, $V(A) = 1$. The set $\textsc{MC}_U \subseteq \mathcal{P}_{\mathsf{fin}}(\mathcal{L}_U) \times \mathcal{L}_U \times \{1, 0, \bot\}$ is defined by: $(\Gamma, A, v) \in \textsc{MC}_U$ iff $V_\Gamma(A) = v$. The particular case for $\Gamma = \emptyset$ is the set denoted $\textsc{MC}^{\emptyset}_U$.

Lemma 9.2. $\textsc{Val}_U$ *is coNP-hard.*

Proof. We show that the *coNP*-complete problem of classical validity $\textsc{Val}$ polynomially reduces to $\textsc{Val}_U$. Let $A \in \mathcal{L}$. For $\mathsf{at}(A) = \{p_1, \ldots, p_n\}$, define: $f(A) = [\pm p_1] \ldots [\pm p_n]A$. Since this function $f : \mathcal{L} \to \mathcal{L}_U$ is linear in $|A|$, it only remains to show that $A \in \textsc{Val}$ iff $f(A) \in \textsc{Val}_U$.
Suppose $A \in \textsc{Val}$. For $A \in \textsc{Val}_U$, it suffices to show that $V_\emptyset(f(A)) = 1$. Fix a choice of literals $\sigma = \{p'_1, \ldots, p'_n\}$ with $p'_i \in \{p_i, \neg p_i\}$. Define a classical assignment by $p_i \mapsto 1$ if $p'_i = p_i$ and $p_i \mapsto 0$ if $p'_i = \neg p_i$, and extend it into a classical interpretation $\mathcal{I}_\sigma : \mathcal{L} \to \{1, 0\}$.

An induction on the complexity of the subformulas $B \in \mathsf{sub}(A)$ shows that $V^\sigma_\emptyset(B) = \mathcal{I}_\sigma(B)$. Indeed, this is trivial for all atoms p_i in A. Then, the classical semantics (contained in f_*) determines a unique value in $\{1, 0\}$ all the way up to A, which is the same for $\mathcal{I}_\sigma$ and $V^\sigma_\emptyset$. Combine this with the assumption $\mathcal{I}_\sigma(A) = 1$ to conclude that $V^\sigma_\emptyset(A) = 1$, for any choice σ.

Applying Definition 7.2 gives $V^{p'_1 \cdots p'_{n-1}}_\emptyset([\pm p_n]A) = 1$ for any choice of $\{p'_1, \ldots, p'_{n-1}\}$ and successive applications render $V_\emptyset([\pm p_1][\pm p_2] \ldots [\pm p_{n-1}]$ $[\pm p_n]A) = 1$. Thus, we proved that $f(A) \in \textsc{Val}_U$.
Suppose now that $A \notin \textsc{Val}$. That is, $\not\models A$. Then since $t(f(A)) = A$, Corollary 8.15 gives (for $\Gamma = \emptyset$) that $\not\models_U f(A)$. Hence, $f(A) \notin \textsc{Val}_U$. ■

Definition 9.3. (*Boolean subformulas; modal path*) For $A \in \mathcal{L}_U$, define the set of *boolean subformulas* of A inductively as follows:

$$\mathsf{bsub}(p) = \{p\} \quad \mathsf{bsub}(\neg A) = \{\neg A\} \cup \mathsf{bsub}(A) \quad \mathsf{bsub}([\pm A]B) = \{[\pm A]B\}$$
$$\mathsf{bsub}(A * B) = \{A * B\} \cup \mathsf{bsub}(A) \cup \mathsf{bsub}(B).$$

```
Push input node to first stack.
Loop while first stack is not empty
    Pop a node from first stack. Apply path.consistency.
    1 If inconsistent, push node to second stack.
    2 Else if node is propositional, apply 0consq. Push node to second stack.
      Loop while first unevaluated node in second stack is preceded by its child
        node(s):
        2.1 evaluate first unevaluated node and delete its child node(s).
    3 Else (path consistent, not an evaluated propositional node) push node to
        second stack. Push left and right children of the node to first stack.
Pop node from second stack. Return node.
```

Figure 10. The post-order depth first search algorithm for model checking a root $(\emptyset, A)$ of a tree $\mathcal{T}$ for $A \in \mathcal{L}_U$. The algorithm takes as input any node of the form (σ, B)

A *modal path* in A is a sequence $\sigma = \langle [\pm A_1]B_1, \dots, [\pm A_m]B_m \rangle$ satisfying $[\pm A_1]B_1 \in \mathsf{bsub}(A)$ and $[\pm A_{i+1}]B_{i+1} \in \mathsf{bsub}([\pm A_i]B_i)$ for $1 \leq i < m$. We abusively call a *modal path* any choice sequence $\sigma = \langle A'_1, \dots, A'_m \rangle \in \langle \pm A_1, \dots, \pm A_m \rangle$. Finally, we refer to the set $\{A'_1, \dots, A'_m\}$ also by σ.

Theorem 9.4. $\mathrm{MC}^{\emptyset}_{U}$ *is in PSpace. Hence,* VAL_U *is also in PSpace.*

Proof. Consider a tree $\mathcal{T} = (N, R)$ of nodes (σ, B) consisting of a modal path $\sigma = \langle A'_1, \dots, A'_m \rangle$ in A and a formula $B \in \mathsf{bsub}([\pm A_m]B_m)$. The edges $R = R_0 \cup R_1$ are split between left nodes (R_0) and right nodes (R_1):

$$\begin{array}{ll} (\sigma, \neg B) R_0 (\sigma, B) & (\sigma, B_0 * B_1) R_i (\sigma, B_i) \\ (\sigma, [\pm A_j]B) R_0 (\sigma.A_j, B) & (\sigma, [\pm A_j]B) R_1 (\sigma.\neg A_j, B) \end{array}$$

where $i \in \{0, 1\}$, $* \in \{\wedge, \vee, \rightarrow\}$. Note the root is $(\emptyset, A)$ and leaf nodes are of the form (σ, p). Our algorithm in Figure 10 traverses (part of) the binary tree $\mathcal{T}$ using post-order traversal and evaluates nodes $(\sigma, B) \mapsto (\sigma, B, v(\sigma, B))$ as either $v(\sigma, B) = V^{\sigma}_{\emptyset}(B)$, or $v(\sigma, B) = \curlywedge$ in case $V^{\sigma}_{\emptyset}$ is 0-depth inconsistent. The algorithm in Figure 10 contains the basic operations:

`path.consistency`(σ, B) adapts the algorithm from D'Agostino [11] testing whether $\sigma \vdash_0 \curlywedge$, and in this case it returns an evaluated node $(\sigma, B, \curlywedge)$.

`0consq`(σ, B) in line (2) is also based on [11] and returns (σ, B, v) according to: $v = 1$ if $\sigma \vdash_0 B$; $v = 0$ if $\sigma \vdash_0 \neg B$ and $v = \bot$ if $\sigma \nvdash_0 B$ and $\sigma \nvdash_0 \neg B$.

`evaluate`$(\sigma, \neg B)$ in line (2.1) returns $(\sigma, \neg B, f_{\neg}(v))$ if the preceding (child) node is (σ, B, v)

`evaluate`$(\sigma, B_0 * B_1)$ in line (2.1) returns $(\sigma, B_0 * B_1, k_*(v_0, v_1))$ if preceded by (σ, B_0, v_0) and (σ, B_1, v_1)[15] where k_* is the Kleene truth-table for $*$.

`evaluate`$(\sigma, [\pm A_m] B_m)$ in line (2.1) computes the restricted pointwise meet (Remark 2) of the values in $(\sigma.A_m, v_0), (\sigma.\neg A_m, v_1)$:

$$meet(v_0, v_1) = \begin{cases} v_0 \sqcap v_1 & \text{if } v_0, v_1 \in \{1, 0, \bot\} \\ 1 & \text{if } v_0 = \curlywedge = v_1 \\ v_i & \text{if } v_i \neq \curlywedge \text{ and } v_{1-i} = \curlywedge. \end{cases}$$

(Correctness and termination.) Note that the algorithm takes any initial node of the form (σ, A). We show by induction on A that $V^{\sigma}_{\emptyset}(A) = v$ iff on input node (σ, A) the algorithm terminates and returns (σ, A, v).

(Base case $A = p$.) A quick inspection of `0consq`(σ, p) (line 2) shows that: $V^{\sigma}_{\emptyset}(p) = v$ iff $V_{\sigma}(p) = v$ iff (σ, p, v) is returned.

(Ind. case $B_0, B_1 \mapsto B_0 * B_1$.) First observe that, by the $\sqsubseteq$-minimality of $V_{\emptyset}$, the only case where $V^{\sigma}_{\emptyset}(B_0 * B_1)$ does not follow the Kleene truth-table $k_{\wedge}(\bot, \bot) = \bot = k_{\vee}(\bot, \bot)$ is precisely when $\sigma \vdash_0 (\neg) B_0 * B_1$ (line 2). Thus, for all other cases, f_* will coincide with k_*, and so $v = V^{\sigma}_{\emptyset}(B_0 * B_1)$ will match the node $(\sigma, B_0 * B_1, v)$ (line 2.1). Let `trace`(σ, A) denote the trace of the algorithm for an input node (σ, A) and let us represent the two stacks in a state by $\langle \alpha \rangle \mid \langle \beta \rangle$ (or by different columns below):

$\langle (\sigma, B_0 * B_1) \rangle$	—	initial state
$\langle (\sigma, B_0), (\sigma, B_1) \rangle$	$\langle (\sigma, B_0 * B_1) \rangle$	line 3
$\langle \alpha, (\sigma, B_1) \rangle$	$\langle \beta, (\sigma, B_0 * B_1) \rangle$	for all $\langle \alpha \rangle \mid \langle \beta \rangle \in$ `trace`(σ, B_0)
$\vdots$	$\vdots$	
$\langle (\sigma, B_1) \rangle$	$\langle (\sigma, B_0, v_0), (\sigma, B_0 * B_1) \rangle$	(ind. hyp. on B_0)
$\langle \gamma \rangle$	$\langle \delta, (\sigma, B_0, v_0), (\sigma, B_0 * B_1) \rangle$	for all $\langle \gamma \rangle \mid \langle \delta \rangle \in$ `trace`(σ, B_1)
$\vdots$	$\vdots$	
—	$\langle (\sigma, B_1, v_1), (\sigma, B_0, v_0),$ $(\sigma, B_0 * B_1) \rangle$	(ind. hyp. on B_1)
—	$\langle (\sigma, B_0 * B_1, v_0 * v_1) \rangle$	returns $(\sigma, B_0 * B_1, v_0 * v_1)$.

[15] In fact, the two child nodes will occur in the reverse order: the second stack will consist of $\langle (\sigma, B_1, v_1), (\sigma, B_0, v_0), (\sigma, B_0 * B_1), \ldots \rangle$.

Thus, the property that *the first unevaluated node in the second stack is preceded by its evaluated child nodes* is preserved from B_0 and B_1 into $B_0 * B_1$. The case for $(\sigma, \neg B_0)$ is even simpler, as it only generates a left child node.

(Ind. case, $[\pm C]B$.) Assume the claim for any modal path σ' and a formula B. Let $\langle\alpha\rangle \mid \langle\beta\rangle$ and $\langle\gamma\rangle \mid \langle\delta\rangle$ denote states in $\texttt{trace}(\sigma.C, B)$ resp. $\texttt{trace}(\sigma.\neg C, B)$

$\langle(\sigma, [\pm C]B)\rangle$	—	initial state
$\langle(\sigma.C, B), (\sigma.\neg C, B)\rangle$	$\langle(\sigma, [\pm C]B)\rangle$	line 3
$\langle\alpha, (\sigma.\neg C, B)\rangle$	$\langle\beta, (\sigma, [\pm C]B)\rangle$	for all $\langle\alpha\rangle \mid \langle\beta\rangle$
$\vdots$	$\vdots$	
$\langle(\sigma.\neg C, B)\rangle$	$\langle(\sigma.C, B, v_0), (\sigma, [\pm C]B)\rangle$	(ind. hyp. on B_0)
$\langle\gamma\rangle$	$\langle\delta, (\sigma.C, B, v_0), (\sigma, [\pm C]B)\rangle$	for all $\langle\gamma\rangle \mid \langle\delta\rangle$
$\vdots$	$\vdots$	
—	$\langle(\sigma.\neg C, B, v_1), (\sigma.C, B, v_0),$ $(\sigma, [\pm C]B)\rangle$	(ind. hyp. on B_1)
—	$\langle(\sigma, [\pm C]B, \mathit{meet}(v_0, v_1))\rangle$	returns this node.

In view of Remark 2, an inspection of *meet* shows that if $V_\emptyset^{\sigma.C}(B) = v_0$ and $V_\emptyset^{\sigma.\neg C}(B) = v_1$ (the ind. hyp.), then $V_\emptyset^{\sigma}([\pm C]B) = \mathit{meet}(v_0, v_1)$.

Thus, in all cases the algorithm terminates and returns a sound input $(\sigma, B, V_\emptyset^{\sigma}(B))$. In particular, for inputs $(\emptyset, A)$ corresponding to the root of the tree $\mathcal{T}$ for A, we obtain that $V_\emptyset(A) = v$ iff the algorithm returns $(\emptyset, A, v)$. (Space complexity) The space complexity of post-order traversal is the depth of the tree, or the depth $d(A)$ of the formula A in our case, which is linear in $|A|$. Our algorithm keeps the same bound by discarding evaluated nodes as soon as they can be used to evaluate a more complex formula (line 2.1).[16] Hence, at any time the two stacks have size polynomial in the length of A, namely $d(A) \cdot |A|$, as for any modal path and its formula $|(\sigma, B)| \leq |A|$. (Time complexity) Observe first that each $\texttt{path.consistency}(\sigma, B)$, each test on $B \in \mathcal{L}$ and each $\texttt{0consq}(\sigma, B)$ operation (lines 1, 2) takes polynomial time $O(|\sigma \cup \{B\}|^2)$ in $|A|$ (by Proposition 1.10); for any other formula, $\texttt{evaluate}(\sigma, B)$ takes constant time (line 2.1). On the other hand, a number of operations exponential in the length of A is required in the worst case (nested modalities): each modal subformula $[\pm A_i]B_i$ doubles the number of required operations upon $\texttt{sub}(B_i)$, namely for the two branches $V_\emptyset^{A_i}$ and

[16] Moreover, the algorithm might not explore the full tree $\mathcal{T}$: at the first sign of 0-depth inconsistency or $\vdash_0$ consequence at a node $(\sigma.A_m, B)$, the subtree below this node is pruned.

$V_\emptyset^{\neg A_i}$. Thus, the number of nodes in the tree is exponential in the length of A. This gives an exponential time upper bound.

In summary, $\mathrm{MC}_U^\emptyset$ is in *PSpace*. We prove that $\textsc{Val}_U$ is also in *PSpace* by providing a polynomial reduction from $\textsc{Val}_U$ to $\mathrm{MC}_U^\emptyset$. This is simply given by the function $A \mapsto (\emptyset, A, 1)$. This is indeed a reduction since $A \in \textsc{Val}_U$ is equivalent to the truth of A in $V_\emptyset$, as the latter is $\sqsubseteq$-minimal among all valuations in $\mathbb{S} \setminus \{V_\curlywedge\}$. ■

In comparison with the general case, the validity problem for proof blueprints under some virtual space function f remains in *coNP*.

Definition 9.5. For a given function f as in Definition 1.8, we define the set:

$$\textsc{bpVal}_U^{\mathsf{f}} = \{[\pm A_1] \ldots [\pm A_k]B \in \textsc{Val}_U : B \in \mathcal{L} \text{ and } A_1, \ldots, A_m \in \mathsf{f}(\{B\})\}.$$

Theorem 9.6. *$\textsc{bpVal}_U^{\mathsf{f}}$ is coNP-complete.*

Proof. ($\textsc{bpVal}_U^{\mathsf{f}}$ is *coNP*-hard.) This can be shown as in the proof that $\textsc{Val}_U$ is *coNP*-hard (Lemma 9.2): the same reduction $f(A)$ gives atomic blueprints and any virtual space function $\mathsf{f}(\cdot)$ contains $\mathsf{at}(\cdot)$ (Definition 1.8). ($\textsc{bpVal}_U^{\mathsf{f}}$ is in *coNP*.) Let $\textsc{Val}_k^{\mathsf{f}}$ denote the set of formulas valid in the degree-bounded logic $\Vdash_k^{\mathsf{f}}$. Recall that for any virtual space function f, $\textsc{Val}_k^{\mathsf{f}}$ is in P (Lemma 6.7). We prove that the linear function $t : [\pm A_1] \ldots [\pm A_k]B \mapsto B$ is a (polynomial) reduction of $\textsc{bpVal}_U^{\mathsf{f}}$ into $\textsc{Val}$.

$$\begin{aligned}
A \in \textsc{bpVal}_U^{\mathsf{f}} \text{ iff } & A = [\pm A_1] \ldots [\pm A_k]B \text{ for some } k \text{ and } B \in \mathcal{L} \text{ and} \\
& \quad \{A_1, \ldots, A_k\} \subseteq \mathsf{f}(\{B\}) \text{ and } \emptyset \models_U [\pm A_1] \ldots [\pm A_k]B \\
\text{iff } & \text{for some } k,\ \emptyset \Vdash_k^{\mathsf{f}} B && \text{(Lemma 8.17)} \\
\text{iff } & \text{for some } k,\ B \in \textsc{Val}_k^{\mathsf{f}} && \text{(by definition)} \\
\text{iff } & B \in \textsc{Val} = \textstyle\bigcup_k \textsc{Val}_k^{\mathsf{f}} && \text{(Lemma 6.5).}
\end{aligned}$$

■

The same argument shows that deciding whether a proof blueprint is a logical consequence of some propositional database Γ is also *coNP*-complete. For any given particular instance, moreover, deciding whether a blueprint $A = [\pm A_1] \ldots [\pm A_k]B$ is valid or a consequence from Γ has polynomial complexity, namely in $O(|B|^{k+1})$ and resp. $O(|\Gamma \cup \{B\}|^{k+1})$. In practice, the actual coefficients will be lower as no choice of *RB* formulas is involved (compare with [11, Alg. 3.1]).

Remark 3. Other decision problems are *satisfiability* and $\bot$-*satisfiability*, denoted $\textsc{Sat}_U$ resp. $\bot\textsc{Sat}_U$. Each is defined as the set of formulas A for which there is some $V \neq V_\curlywedge$ with $V(A) = 1$ and resp. $V(A) = \bot$. While $\textsc{Sat}_U$ can be shown to be *NP*-hard using the same function f as in Lemma 9.2, $\bot\textsc{Sat}_U$ is in *PSpace* (by Theorem 9.4 and that $A \in \bot\textsc{Sat}_U$ iff $V_\emptyset(A) = \bot$.)

Applications in Distributed Reasoning As argued in the Introduction, the form of deductive exchanges (messages) among a network of reasoning agents should satisfy certain desiderata, including: quick correctness tests, optimality preservation and succintness. For proof blueprints, generated directly from $\vdash_U$ or encoding some $\Vdash^{\mathsf{f}}_k$-proof, these properties are granted by: the discussion following Theorem 9.6 above, the proof normalization procedures for resource-bounded logics [13] and the fact that any proof blueprint for $B \in \mathcal{L}$ is linear in the length of B. Two further advantages of proof blueprints are described next.

Example 11. (*Distributed reasoning*) Assume a set of resource-bounded agents share a propositional database Γ, and are tasked with answering queries. Say an agent, executing the algorithm for some degree-bounded logic $\Vdash^{\mathsf{f}}_k$, receives a message containing a formula B, that it might use as a lemma towards the query. This agent cannot know if it can verify that $\Gamma \vdash B$ as a negative answer $\Gamma \nVdash^{\mathsf{f}}_k B$ does not rule out that $\Gamma \Vdash^{\mathsf{f}}_n B$ for some $n > k$.

If, on the contrary, the message is a proof blueprint $[\pm A_1], \ldots, [\pm A_m]B$, then it can easily check whether $m \leq k$ and $\{A_1, \ldots, A_m\} \subseteq \mathsf{f}(\{B\})$ in which case it instantly knows that it can verify such a claim.

A second advantage of proof blueprints $[\pm A_1], \ldots, [\pm A_m]B$ lies in the existence of syntactic transformations upon them that preserve soundness (or classical provability of their propositional matrix B), such as:

(*merge*) a pair of blueprints $[\pm A] \ldots [\pm A']B$ and $[\pm C] \ldots [\pm C']D$ can be merged into $[\pm A] \ldots [\pm A'][\pm C] \ldots [\pm C']B \wedge D$ as a generalization of the modal intelim rule $I[\pm]\wedge$ (derived using *Triv* and *Perm*); and analogously for generalizations of any modal intelim rule $I[\pm]*$ or $E[\pm]*$;

(*optimize*) an agent can optimize a blueprint $[\pm A] \ldots [\pm A']B$ by removing any repeated modalities (using permutation and contraction) or modalities with an *RE*-equivalent update in the prefix (using *RE*);

(*cut*) suppose the agent proves $[\pm C_1] \ldots [\pm C_k]A'_i$ where $A'_i \in \{A_i, \neg A_i\}$ is a *RB* assumption of a blueprint $[\pm A_1] \ldots [\pm A_k]B$. Then it can replace $[\pm A_i]$ by $[\pm C_1] \ldots [\pm C_k]$ in the latter blueprint (using $E[\pm]$).

These operations facilitate the management of large collections of proof blueprints towards query solving from a propositional data base, or consistency checks upon it. In summary, an exchange format consisting of blueprints (and the actual premises used in a proof) fulfils the above desiderata and can speed up distributed proof and refutation methods.

Conclusions

The informational 3-valued semantics and natural deduction system studied in [10,11,13] was adopted to define a new hierarchy of tractable logics UBBL that approximate classical logic. These logics bound the number of formulas discharged in proofs by the rule of bivalence (RB): if B follows from A and from $\neg A$, then B. In a lattice of valuations, we showed that RB can be simulated by updates $V \mapsto V^{[\pm]A}$ resulting in valuations that behave classically w.r.t. A and thus validate $A \vee \neg A$. We also proved that updates satisfy permutation and reduce to theory expansions.

A modal logic $\models_U$ was then presented where formulas $[\pm A] \dots [\pm A']B$ make explicit RB-instances that would suffice in a classical proof of B. The logic thus contains all classical validities prefixed with updates, such as $[\pm p](p \vee \neg p)$, and is a conservative extension of the RB-free fragment of classical logic. A natural deduction system $\vdash_U$ with introduction and elimination rules for $[\pm A]$ and reduction rules for negation was proved to be sound and complete for $\models_U$. We also identified derivable rules which permit a comparison with modal axioms and natural deduction systems for modal logics [21,29]. The computational complexity of its validity problem was also studied, setting a *PSpace* upper bound. Towards applications in distributed reasoning, we established the complexity class for validity over proof blueprints as *coNP*-complete, thus improving on the complexity of the (general) validity problem. Proof blueprints are thus a robust, computationally feasible message form for deductive exchanges in distributed reasoning. A number of syntactic operations on proof blueprints enable quick management techniques for merging and optimizing such proof blueprints.

As for future work, an interesting question is whether the natural deduction system $\vdash_U$ enjoys some form of the subformula property and proof normalization; for DBBL this has been shown to be the case for the weak subformula property [13]. Also left as an open question is a sharp characterization of the complexity of validity for $\vdash_U$.

In relation to DBBL, one might also ask what modal logic of updates can capture the DBBL hierarchy, as $\models_U$ does for UBBL. To this end, instead of representing RB applications in a proof as a sequence $[\pm A] \ldots [\pm A']$, the depth-bounded case might use conditional progams π built from: updates $\pm C$, tests $?C$, composition $\pi;\pi'$ and choice $\pi \cup \pi'$. We conjecture that a PDL-style language [20] with formulas $[\pi]A$ built from these programs, the formula $[\pm A; (?A;\pi) \cup (?\neg A;\pi')]B$ would express that B is a $(k{+}1)$-depth consequence if π and π' are k-depth conditional programs.

Towards distributed reasoning applications, a further step would be to extend the modal logic $\models_U$ with modalities for public announcements of proof blueprints, with formulas $\langle ![\pm A_1] \ldots [\pm A_k]B \rangle C$ expressing: C is known after $[\pm A_1] \ldots [\pm A_k]B$ is truthfully announced.

Finally, a general open question is how arbitrary modal logics build upon resource-bounded logics, rather than classical logic. Of particular interest would be epistemic logics addressing the logical omniscience problem, but the question is equally pertinent to all modal logics starting from K, as studied in [17] for a tableaux system.

Acknowledgements. The author wishes to thank the anonymous referees for their many interesting comments and corrections. This research was funded by the Department of Philosophy "Piero Martinetti" of the University of Milan under the project "Departments of Excellence 2018-2022" awarded by the Ministry of Education, University and Research (MIUR).

Funding Open access funding provided by Università degli Studi di Milano within the CRUI-CARE Agreement.

A. Appendix: Lattice theory

Basic concepts and results for lattices are listed next, taken from Davey and Priestley [16] or Grätzer [24].

Definition A.1. (*Poset*) A partially ordered set, or *poset*, is a pair $(X, \leq)$ where the relation $\leq \subseteq X \times X$ satisfies: (reflexivity) $x \leq x$, (antisymmetry) $x \leq y \leq x$ implies $x = y$, and (transitivity) $x \leq y \leq z$ implies $x \leq z$.

Definition A.2. (*Lower, upper bound; meet, join; minimum*) Given a poset $(X, \leq)$, a subset $Y \subseteq X$, we say that $\alpha \in X$ is

- a *lower bound of* Y if $\alpha \leq y$ for all $y \in Y$; the set of lower bounds of Y is denoted Y^ℓ;
- the *meet* of Y, denoted $\bigsqcap Y$, if α is the greatest lower bound: $\beta \in Y^\ell$ implies $\beta \leq \alpha$. If moreover $\bigsqcap Y \in Y$, it is called the *minimum* of Y.

Reversing the order into $\geq$ gives the dual notions of: *upper bound*; the set Y^u of upper bounds of Y; and the *join* $\bigsqcup Y$ or least upper bound of Y. In particular, $\bigsqcap \emptyset$ is the $\leq$-maximum and $\bigsqcup \emptyset$ the $\leq$-minimum element.

Definition A.3. (*Semi- and complete lattice*) A *meet semi-lattice* $(X, \sqsubseteq)$ is a poset closed under meets for non-empty subsets: $\bigsqcap Y \in X$ whenever $\emptyset \neq Y \subseteq X$. A *complete lattice* is a poset $(X, \leq)$ closed under arbitrary meets and joins: $Y \subseteq X$ implies $\bigsqcap Y, \bigsqcup Y \in X$.

Lemma A.4. *(*[16, p.57], Definition 4.6*) In any lattice* $(X, \leq)$, $\uparrow(x \sqcup y) = \uparrow x \cap \uparrow y$.

Lemma A.5. (Order-preserving [16, p.35]) *In any lattice* $(X, \leq)$,

$$x \sqcap y \leq x, y \leq x \sqcup y \qquad\qquad x \leq y \text{ iff} \uparrow y \subseteq \uparrow x$$

Both $\sqcap$ *and* $\sqcup$ *are order-preserving:* $x \leq y$ *and* $x' \leq y'$ *implies* $x \sqcap x' \leq y \sqcap y'$*; in particular,* $x, x' \leq y$ *implies* $x \sqcap x' \leq y$*. And the same holds for* $\sqcup$*.*

Lemma A.6. (Connecting lemma [16, p.390]) *For any lattice* $(X, \leq)$,

$$x \sqcap y = x \quad \text{iff} \quad x \leq y \quad \text{iff} \quad x \sqcup y = y.$$

Theorem A.7. (Join and meet [16, p. 39]) *For any lattice* $(X, \leq)$*, its join* $\sqcup$ *is associative, commutative , idempotent and satisfies the absorption law:*

$$\begin{array}{ll} x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z \text{ (assoc.)} & x \sqcup y = y \sqcup x \quad \text{(comm.)} \\ x \sqcup x = x \qquad\qquad \text{(idemp.)} & x \sqcap (y \sqcup x) = x \text{ (absorption)} \end{array}$$

and dual laws hold for meet $\sqcap$ *(i.e. by switching* $\sqcap$ *and* $\sqcup$ *above).*

Lemma A.8. (Semi-distributivity [16, Lemma 4.1], [24, p. 14]) *For any lattice $(X, \leq)$ and arbitrary elements $x, y, z \in X$, it holds that:*

$$x \wedge (y \vee z) \geq (x \wedge y) \vee (x \wedge z) \quad \text{and} \quad x \vee (y \wedge z) \leq (x \vee y) \wedge (x \vee z).$$

B. Appendix: Proofs

Proposition **3.4**. For any non-empty subset $\mathbb{V}' \subseteq \mathbb{V}$, the meet exists in $\mathbb{V}$ as the pointwise meet $\bigsqcap \mathbb{V}' = \{\langle A, \bigsqcap_{V \in \mathbb{V}'} V(A)\rangle : A \in \mathcal{L}\}$.

Proof. The pointwise meet of a set of 3-valued functions is clearly its greatest lower bound. Let us show then $\bigsqcap \mathbb{V}' \in \mathbb{V}$, i.e. that $\bigsqcap \mathbb{V}'$ respects the truth-tables. For negation $\neg$, we omit *for all* $V \in \mathbb{V}'$ and reason as follows:

$$(\bigsqcap \mathbb{V}')(\neg A) = \begin{cases} 1 & \text{if } V(\neg A) = 1 \Leftrightarrow \text{if } V(A) = 0 \Leftrightarrow (\bigsqcap \mathbb{V}')(A) = 0 \\ 0 & \text{if } V(\neg A) = 0 \Leftrightarrow \text{if } V(A) = 1 \Leftrightarrow (\bigsqcap \mathbb{V}')(A) = 1 \\ \bot & \text{other cases} \Leftrightarrow \text{other cases} \Leftrightarrow (\bigsqcap \mathbb{V}')(A) = \bot \end{cases}$$

where other cases $X \subseteq \{1, 0, \bot\}$ in the last $\Leftrightarrow$ step amount to $\{0\} \neq X \neq \{1\}$ and so $\bot = \bigsqcap X$. For the remaining connectives, we only prove the conjunction case by considering all possible values $\langle v, v'\rangle = \langle (\bigsqcap \mathbb{V}')(A), (\bigsqcap \mathbb{V}')(B)\rangle$:

$\langle 0, v'\rangle; \langle v, 0\rangle$. We only check $v = (\bigsqcap \mathbb{V}')(A) = 0$. Then clearly $V(A) = 0$ for all $V \in \mathbb{V}'$ and so $V(A \wedge B) = 0$ for all $V \in \mathbb{V}'$. Hence $(\bigsqcap \mathbb{V}')(A \wedge B) = 0$.

$\langle 1, 1\rangle$. In this case, $V(A \wedge B) = 1$ for all $V \in \mathbb{V}'$ and so $(\bigsqcap \mathbb{V}')(A \wedge B) = 1$.

$\langle 1, \bot\rangle; \langle \bot, 1\rangle$. Say $(\bigsqcap \mathbb{V}')(B) = \bot$. Then the set of B-values $X = \{V(B) : V \in \mathbb{V}\}$ satisfies: $\{1\} \neq X \neq \{0\}$ and so $\bigsqcap X = \bot$. Since $f_\wedge(1, v') = v'$, we also have $X = \{V(A \wedge B) : V \in \mathbb{V}'\}$ which gives $(\bigsqcap \mathbb{V}')(A \wedge B) = \bigsqcap X = \bot$.

$\langle \bot, \bot\rangle$. We check that $(\bigsqcap \mathbb{V}')(A \wedge B) \in \{0, \bot\}$. Let X, Y be the sets of values of A resp. B among $\mathbb{V}'$. Again each set is different from $\{1\}$ and $\{0\}$, so if $(\bigsqcap \mathbb{V}')(A \wedge B) = 1$ the sets would be $X = \{1\} = Y$ (contradiction). ■

Fact **3.8**. $\mathbb{V}^+$ is the set of functions $V : \mathcal{L} \to \{1, 0, \bot\}$ closed under applications of intelim rules.

Proof. Given Proposition 3.2, the $(\subseteq)$ direction reduces to checking that $V_{\curlywedge}$ is closed under intelim rules, and indeed all rule conclusions are true in $V_{\curlywedge}$. For $(\supseteq)$, let f' be in the set $\{f : \mathcal{L} \to \{1, 0, \bot\} \mid f \text{ is closed under intelim rules}\}$. In case f' respects the truth-tables, by Definitions 1.2, 3.1 we obtain that $f' \in \mathbb{V}$ and we are done. In case f' does not respect some truth-table, a routine examination of all cases shows that either a contradiction occurs of the form $f'(A) = 1$ and $f'(\neg A) = 1$, or that $(A, v), (A, v') \in f'$ with $v \neq v'$. In the first case, we must have $f' = V_{\curlywedge}$ since closure under $I{\curlywedge}$ and $E{\curlywedge}$ implies

$V(B) = 1$ for any B; in the second case, we arrive at a contradiction with the assumption that f' is a function. Hence, $f' \in V_{\curlywedge}$. ∎

Fact **4.2**. For any $V \in \mathbb{V}^+$, $\mathsf{Th}(V)$ is a theory: $\mathsf{Cn}_0(\mathsf{Th}(V)) \subseteq \mathsf{Th}(V)$.

Proof. From $A \in \mathsf{Cn}_0(\mathsf{Th}(V))$, we get $\mathsf{Th}(V) \vdash_0 A$, so by soundness (Proposition 1.11) $\mathsf{Th}(V) \models_0 A$. In particular, if $V[\mathsf{Th}(V)] = \{1\}$ then $V(A) = 1$. But the antecendent is immediate, so $V(A) = 1$ and thus $A \in \mathsf{Th}(V)$. ∎

Lemma **4.3**. For any $V, V' \in \mathbb{V}^+$, $\mathsf{Th}(V) \subseteq \mathsf{Th}(V')$ iff $V \sqsubseteq V'$.

Proof. ($\Leftarrow$.) The assumption gives $V(A) \leq V'(A)$, so if $V(A) = 1$, we also have $V'(A) = 1$. Hence $\mathsf{Th}(V) \subseteq \mathsf{Th}(V')$. ($\Rightarrow$.) Assume $\mathsf{Th}(V) \subseteq \mathsf{Th}(V')$ and let $A \in \mathcal{L}$. We show $V(A) \leq V'(A)$. (Case $V(A) = 1$.) This is simply $A \in \mathsf{Th}(V)$ and so the assumption gives $A \in \mathsf{Th}(V')$ and finally $V'(A) = 1$. (Case $V(A) = 0$.) Then $V(\neg A) = 1$ and reasoning as in the previous case we obtain $V'(\neg A) = 1$ so $V'(A) = 0$. Again $V(A) \leq V'(A)$. (Case $V(A) = \bot$.) Clearly, $\bot \leq V'(A)$. These cases show $V(A) \leq V'(A)$ and so $V \sqsubseteq V'$. ∎

Proposition **4.5**. For each set $\Gamma \subseteq \mathcal{L}$, a $\sqsubseteq$-minimum valuation V_Γ exists in $\mathbb{V}_\Gamma$. As a consequence $V_\Gamma = \bigsqcap \mathbb{V}_\Gamma$.

Proof. In case Γ is $\vdash_0$-inconsistent, $\mathbb{V}_\Gamma = \{V_{\curlywedge}\}$, so $\bigsqcap \mathbb{V}_\Gamma = V_{\curlywedge} \in \mathbb{V}_\Gamma$ is the $\sqsubseteq$-minimum. For an $\vdash_0$-consistent set Γ, we inductively define V_Γ via partial maps. Let $V^0(A) = 1$ for each $A \in \Gamma$ and $V^0(A) = \mathit{und.}$ (*undedfined*) otherwise. Let V^{n+1} be defined by: $V^n \subseteq V^{n+1}$ and for $A \in \mathcal{L} \setminus \mathrm{dom}(V^n)$,

$$V^{n+1}(A) = \begin{cases} 1 & \text{for an intelim rule } B, C \therefore A, V^n(B) = 1 = V^n(C) \\ 0 & \text{if } V^n(\neg A) = 1 \\ \mathit{und.} & \text{otherwise} \end{cases}$$

Let $V^\omega = \bigcup_{n<\omega} V^n$ and finally define $V_\Gamma = V^\omega \cup \{\langle A, \bot \rangle : A \in \mathcal{L} \setminus \mathrm{dom}(V^\omega)\}$.

Since Γ is 0-depth consistent, an easy induction shows that no V^n contains an inconsistent pair $\langle A, 1\rangle, \langle A, 0\rangle$ and so each V^n is a function $V^n : \mathcal{L} \to \{1, 0\}$. Hence the same holds for V^ω. This and the fact that V_Γ is closed under intelim rules imply that V_Γ satisfies the truth-tables (Fact 3.2) and so $V_\Gamma \in \mathbb{V}$. Vice versa, for any $V \in \mathbb{V}_\Gamma$, $V_\Gamma(A) = 1$ implies $V(A) = 1$, again by Fact 3.2 and $V[\Gamma] = \{1\}$. That is, $\mathsf{Th}(V_\Gamma) \subseteq \mathsf{Th}(V)$ so by Lemma 4.3, $V_\Gamma \sqsubseteq V$ and moreover $V_\Gamma \sqsubseteq \bigsqcap \mathbb{V}_\Gamma$. With $V_\Gamma \in \mathbb{V}_\Gamma$, we conclude $V_\Gamma = \bigsqcap \mathbb{V}_\Gamma$. ∎

Fact **4.7**. For any set $\Gamma \subseteq \mathcal{L}$, (i) $\mathbb{V}_\Gamma = {\uparrow} V_\Gamma$ and (ii) $\bigsqcap {\uparrow} V_\Gamma = V_\Gamma$.

PROOF. For (i) let $V' \in \mathbb{V}_\Gamma$ be arbitrary. Then,

$$\text{(i)}\quad \begin{aligned} V' \in \mathbb{V}_\Gamma &\text{ iff } V'[\Gamma] = \{1\} \\ &\text{ iff } V_\Gamma \sqsubseteq V' \in \mathbb{V}^+ \\ &\text{ iff } V' \in \uparrow V_\Gamma \end{aligned} \qquad \text{(ii)}\quad \begin{aligned} V_\Gamma &= \textstyle\bigsqcap \mathbb{V}_\Gamma \ \ \text{(Prop. 4.5)} \\ &= \textstyle\bigsqcap \uparrow V_\Gamma \text{ (by (i)).} \end{aligned}$$

■

COROLLARY **4.10**. For any $\Gamma \subseteq \mathcal{L}$, (i) $V_\Gamma = V_{\mathsf{Cn}_0(\Gamma)}$ and (ii) $\mathsf{Cn}_0(\Gamma) = \mathsf{Th}(V_\Gamma)$.

PROOF. For claim (i). ($\sqsubseteq$) $V_\Gamma \sqsubseteq V_{\mathsf{Cn}_0(\Gamma)}$ follows from $\Gamma \subseteq \mathsf{Cn}_0(\Gamma)$ by reasoning similarly to the proof of Lemma 4.3($\Rightarrow$). ($\sqsupseteq$) Let $A \in \mathcal{L}$ be arbitrary with $V_{\mathsf{Cn}_0(\Gamma)}(A) = 1$. That is, $\mathsf{Cn}_0(\Gamma) \models_0 A$, and so by completeness $\mathsf{Cn}_0(\Gamma) \vdash_0 A$. Since $\mathsf{Cn}_0(\Gamma)$ a theory, $A \in \mathsf{Cn}_0(\Gamma)$ and hence $\Gamma \vdash_0 A$. Now by soundness $\Gamma \models_0 A$ and finally we obtain $V_\Gamma(A) = 1$ using Corollary 4.8. This gives $\mathsf{Th}(V_{\mathsf{Cn}_0(\Gamma)}) \subseteq \mathsf{Th}(V_\Gamma)$ so Lemma 4.3 gives $V_{\mathsf{Cn}_0(\Gamma)} \sqsubseteq V_\Gamma$. For claim (ii),

$$\begin{aligned} \mathsf{Cn}_0(\Gamma) &= \mathsf{Th}(V_{\mathsf{Cn}_0(\Gamma)}) \text{ by Prop. 4.9}(i) \\ &= \mathsf{Th}(V_\Gamma) \text{ by (i) with } V_{\mathsf{Cn}_0(\Gamma)} = V_\Gamma. \end{aligned}$$

■

FACT **4.11**. $\mathsf{Th}(V_1 \sqcap V_2) = \mathsf{Th}(V_1) \cap \mathsf{Th}(V_2)$. As a consequence, $V_\Gamma \sqcap V_\Delta = V_{\mathsf{Cn}_0(\Gamma) \cap \mathsf{Cn}_0(\Delta)}$.

PROOF. We reason as follows:

$$\begin{aligned} \mathsf{Th}(V_1 \sqcap V_2) &= \{A \in \mathcal{L} : \big(V_1 \sqcap V_2\big)(A) = 1\} \\ &= \{A \in \mathcal{L} : V_1(A) = 1\} \cap \{A \in \mathcal{L} : V_2(A) = 1\} \\ &= \mathsf{Th}(V_1) \cap \mathsf{Th}(V_2). \\ \mathsf{Th}(V_\Gamma \sqcap V_\Delta) &= \mathsf{Th}(V_\Gamma) \cap \mathsf{Th}(V_\Delta) \qquad \text{(by the above claim)} \\ &= \mathsf{Cn}_0(\Gamma) \cap \mathsf{Cn}_0(\Delta) \qquad \text{(by Fact 4.7}(i)). \end{aligned}$$

This and Proposition 4.9(i) imply that $V_\Gamma \sqcap V_\Delta = V_{\mathsf{Th}(V_\Gamma \sqcap V_\Delta)} = V_{\mathsf{Cn}_0(\Gamma) \cap \mathsf{Cn}_0(\Delta)}$.

■

PROPOSITION **4.12**. The join $\bigsqcup \mathbb{V}' = \bigsqcap (\mathbb{V}')^u$ obtained from Proposition 3.7 is: $\bigsqcup \mathbb{V}' = V_{\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')}$. Hence, $V_\Gamma \sqcup V_\Delta = V_{\Gamma \cup \Delta}$.

PROOF. The following identities result from applying Lemma 4.3 twice:

$$\begin{aligned} (\mathbb{V}')^u &= \{V^\star : V^\star \sqsupseteq V', \text{for all } V' \in \mathbb{V}'\} \\ &= \{V^\star : \mathsf{Th}(V^\star) \supseteq \mathsf{Th}(V'), \text{for all } V' \in \mathbb{V}'\} \\ &= \{V^\star : \mathsf{Th}(V^\star) \supseteq \textstyle\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')\} \\ &= \{V^\star : V^\star \sqsupseteq V_{\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')}\} \ = \ \uparrow V_{\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')}. \end{aligned}$$

Hence $\bigsqcap (\mathbb{V}')^u(A) = 1$ iff $\bigsqcap \uparrow V_{\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')}(A) = 1$ iff $V_{\bigcup_{V' \in \mathbb{V}'} \mathsf{Th}(V')}(A) = 1$. That is, $\mathsf{Th}(\bigsqcap (\mathbb{V}')^u) = \mathsf{Th}(V_{\bigcup_{V' \in \mathbb{V}} \mathsf{Th}(V')})$ and so another application of

Lemma 4.3 gives $\bigsqcap(\mathbb{V}')^u = V_{\bigcup_{V'\in\mathbb{V}}\mathsf{Th}(V')}$. For the claim, $V_\Gamma \sqcup V_\Delta = V_{\Gamma\cup\Delta}$,

$$\begin{aligned} V_\Gamma \sqcup V_\Delta = \bigsqcup\{V_\Gamma, V_\Delta\} &= V_{\mathsf{Th}(V_\Gamma)\cup\mathsf{Th}(V_\Delta)} && \text{(by the above claim)}\\ &= V_{\mathsf{Cn}_0(\Gamma)\cup\mathsf{Cn}_0(\Delta)} && \text{(by Coro. 4.10}(ii))\\ &= V_{\mathsf{Cn}_0(\mathsf{Cn}_0(\Gamma)\cup\mathsf{Cn}_0(\Delta))} && \text{(by Coro. 4.10}(i))\\ &= V_{\mathsf{Cn}_0(\Gamma\cup\Delta)} && \text{(since } \vdash_0 \text{ is tarskian)}\\ &= V_{\Gamma\cup\Delta} && \text{(by Coro. 4.10}(i)). \end{aligned}$$ ■

Corollary **5.3**. (i) If $V(A) = 1$, then $V^A = V$; (ii) if $V(A) = 0$, then $V^A = V_\Yup$.

Proof. For (i), if either $V \in \mathbb{V}$ or $V = V_\Yup$, $V(A) = 1$ implies $V_{\{A\}} \sqsubseteq V$. By Lemma A.6, $V \sqcup V_{\{A\}} = V$, so $V^A = V$. For (ii), $V(A) = 0$ and $V_{\{A\}}(A) = 1$ imply $\uparrow V^A = \uparrow V \cap \uparrow V_{\{A\}} = \{V_\Yup\}$ and then $V^A = \bigsqcap \uparrow V^A = \bigsqcap\{V_\Yup\} = V_\Yup$. ■

Fact **5.6**. If $V(A) \neq \bot$, then $V^{[\pm]A} = V$.

Proof. By Corollary 5.3, if $V(A) = 1$ resp. $V(A) = 0$, then $V^{[\pm]A} = V \sqcap V_\Yup$ resp. $V^{[\pm]A} = V_\Yup \sqcap V$. By Lemma A.6 and $V \sqsubseteq V_\Yup$ we get $V \sqcap V_\Yup = V = V_\Yup \sqcap V$, so in any case we obtain $V^{[\pm]A} = V$. ■

Proposition **5.7**. For any virtual space function f (Definition 1.8), $\Gamma \models^{\mathsf{f}}_1 B$ iff $V_\Gamma^{[\pm]A}(B) = 1$ for some $A \in \mathsf{f}(\Gamma \cup \{B\})$.

Proof. We use Lemma 5.4 and Corollary 4.8. Let $A \in \mathsf{f}(\Gamma \cup \{B\})$. Then,

$$\begin{aligned} &V_\Gamma^{[\pm]A}(B) = 1 \;\Leftrightarrow\; (V_\Gamma^A \wedge V_\Gamma^{\neg A})(B) = 1 \;\Leftrightarrow\; V_\Gamma^A(B) = 1 = V_\Gamma^{\neg A}(B)\\ &\Leftrightarrow\; V_{\Gamma\cup\{A\}}(B) = 1 = V_{\Gamma\cup\{\neg A\}}(B) \;\Leftrightarrow\; \Gamma \cup \{\pm A\} \models_0 B \;\Leftrightarrow\; \Gamma \models^{\mathsf{f}}_1 B. \end{aligned}$$

■

Fact **6.4**. It holds that $\Vdash_1 \;=\; \models_1$ and $\Vdash_k \subsetneq \models_k$ for any $k > 1$.

Proof. The claim $\Vdash_1 = \models_1$ follows from Proposition 5.7. The inclusion $\Vdash_k \subseteq \models_k$ is a consequence of Proposition 2.2. This inclusion is proper $\Vdash_k \subsetneq \models_k$, as shown by a generalization of the example Table 1(mid) from $k = 2$ to any k. ■

Proposition **6.5**. $\models \;=\; \bigcup_k \Vdash^{\mathsf{f}}_k$, for any f as in Definition 1.8.

Proof. Assume $\Gamma \models B$. By compactness there is a finite $\Delta \subseteq \Gamma$ with $\Delta \models B$. Applying [10, Prop. 3.2] gives us that $\Delta \models^{\mathsf{f}}_k B$ for some k, so by Proposition 2.2 we obtain $\Delta \Vdash^{\mathsf{f}}_{2^k-1} B$ and so $\Gamma \Vdash^{\mathsf{f}}_{2^k-1} B$. ■

Proposition **6.6**. For any $\Gamma \cup \{A, B\} \subseteq \mathcal{L}$ with $A \in \mathsf{f}(\Gamma \cup \{B\})$, if $\Gamma \Vdash^{\mathsf{f}}_i A$ and $\Gamma \cup \{A\} \Vdash^{\mathsf{f}}_j B$ then $\Gamma \Vdash^{\mathsf{f}}_{i+j} B$.

Proof. From the assumptions, there are $C_1, \dots, C_i \in \mathsf{f}(\Gamma \cup \{A\})$ and $C_{i+1}, \dots, C_{i+j} \in \mathsf{f}(\Gamma \cup \{A, B\})$ with

$$\Gamma \cup \{\pm C_1, \dots, \pm C_i\} \models_0 A \quad \text{and} \quad \Gamma \cup \{A\} \cup \{\pm C_{i+1}, \dots, \pm C_{i+j}\} \models_0 B.$$

Hence, $\Gamma \cup \{\pm C_1, \dots, \pm C_i, \pm C_{i+1}, \dots, \pm C_{i+j}\} \models_0 A$ since the first i choices in this set already imply A. All these C-formulas are also in $\mathsf{f}(\Gamma \cup \{A, B\})$ and so the assumption $A \in \mathsf{f}(\Gamma \cup \{B\})$ gives the first inclusion:

$$\mathsf{f}(\Gamma \cup \{A, B\}) \subseteq \mathsf{f}(\Gamma \cup \mathsf{f}(\Gamma \cup \{B\}) \cup \{B\}) \subseteq \mathsf{f}(\Gamma \cup \{B\})$$

while the second inclusion follows from Definition 1.8(ii). Thus, all the C-formulas are in $\mathsf{f}(\Gamma \cup \{B\})$, and so we are done: $\Gamma \Vdash^{\mathsf{f}}_{i+j} B$. ■

Lemma **6.9**. For any $V \in \mathbb{V}$, the following are equivalent:

(i) $V \not\sqsubseteq \overline{V}$, for any $\overline{V} \in \mathbb{C}$;

(ii) $V^{[\pm]A_1 \cdots [\pm]A_k} = V_{\curlywedge}$, for some $A_1, \dots, A_k \in \mathcal{L}$;

(iii) $\mathrm{V}(\mathrm{A}) = 1$ for some classical inconsistency A, e.g. $\vdash \neg$ A.

Proof. (ii) ⇒ (i). Suppose $V^{[\pm]A_1 \cdots [\pm]A_k} = V_{\curlywedge}$ and, towards a contradiction, that $V \sqsubseteq \overline{V}$ for some $\overline{V} \in \mathbb{C}$. Since $\overline{V}(A_1) \neq \bot$, using Fact 5.6 we get $\overline{V}^{[\pm]A_1} = \overline{V}$ and, similarly, $\overline{V}^{[\pm]A_1 \dots [\pm]A_k} = \overline{V}$. With this and the assumptions, we obtain $V_{\curlywedge} = V^{[\pm]A_1 \cdots [\pm]A_k} \sqsubseteq \overline{V}^{[\pm]A_1 \cdots [\pm]A_k} = \overline{V}$ (contradiction).
(iii) ⇒ (ii). Assume $V(A) = 1$ with $\vdash \neg A$ and let $\mathsf{at}(\neg A) = \{p, \dots, p'\}$.

$$\begin{aligned} \models \neg A \Rightarrow \Vdash^{\mathsf{at}}_{k} \neg A \qquad & \text{for some } k \leq |\{p, \dots, p'\}| \qquad && \text{(Prop. 6.5)} \\ \Rightarrow V'^{[\pm]p_1 \dots [\pm]p_k}(\neg A) = 1 \qquad & \text{for any } V' \in \mathbb{V} && \text{(Def. 7.2)} \end{aligned}$$

In particular, $V^{[\pm]p_1 \dots [\pm]p_k}(\neg A) = 1$ and since $V \sqsubseteq V^{[\pm]p_1 \dots [\pm]p_k}$, the assumption $V(A) = 1$ implies $V^{[\pm]p_1 \dots [\pm]p_k}(A) = 1$. Using Fact 3.8, we obtain $V^{[\pm]p_1 \dots [\pm]p_k}(B) = 1$ for any $B \in \mathcal{L}$. That is, $V^{[\pm]p_1 \dots [\pm]p_k} = V_{\curlywedge}$.
(i) ⇒ (iii). We prove the contrapositive. Assume that for each classical inconsistency $A \in \mathcal{L}$, $V(A) \in \{\bot, 0\}$. For an enumeration $\langle A_n \rangle_{n \in \mathbb{N}}$ of $\mathcal{L}$, we extend V into a two-valued $\overline{V} \in \mathbb{C}$ function. Define $V^0 = V$ and

$$V^{n+1} = \begin{cases} V^n & \text{if } V(A_{n+1}) \neq \bot \text{ or } \vdash \neg A_{n+1} \\ V_{\mathsf{Th}(V^n) \cup \{A_{n+1}\}} & \text{otherwise} \end{cases}$$

The inductive step $V^n \mapsto V^{n+1}$ clearly preserves classical consistency and membership in $\mathbb{V}$. As a consequence, also $V^{\omega} = \bigsqcup_n V^n$ is classically consistent ($V^{\omega}(A) \neq 1$ whenever $\vdash \neg A$) and a valuation in $\mathbb{V}$. The latter, together with $V(p_i) \in \{1, 0\}$ for any $p_i \in \mathsf{Var}$, implies that $V \sqsubseteq V^{\omega} \in \mathbb{C}$. ■

References

[1] Ågotnes, T., and M. Walicki, A logic of reasoning, communication and cooperation with syntactic knowledge, in *4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, Association for Computing Machinery, New York, 2005, pp. 1135–1136.

[2] Alechina, N., B. Logan, and M. Whitsey, A complete and decidable logic for resourcebounded agents, in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems: Volume 2 (AAMAS 2004)*, IEEE Computer Society, New York, 2004, pp. 606–613.

[3] Artemov, S.N., and E. Nogina, Introducing justification into epistemic logic, *Journal of Logic and Computation* 15(6):1059–1073, 2005.

[4] Bendall, K., Natural deduction, separation and the meaning of logical operators, *Journal of Philosophical Logic* 7(1):245–276, 1978.

[5] Bjerring, J.Ch., and M. Skipper, A dynamic solution to the problem of logical omniscience, *Journal of Philosophical Logic* 48(3):501–521, 2019.

[6] Buss, S. R., Polynomial size proofs of the propositional pigeonhole principle, *Journal of Symbolic Logic* 52(4):916–927, 1987.

[7] Cignarale, G., and G. Primiero, A multi-agent depth bounded boolean logic, in L. Cleophas, and M. Massink, (eds.), *Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops*, Springer, 2021, pp 176–191.

[8] Cook, S.A., The complexity of theorem-proving procedures, in M.A. Harrison, N.B. Banerji, and J.D. Ullman, (eds.), *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, 1971, pp. 151–158.

[9] D'Agostino, M., Semantic information and the trivialization of logic: Floridi on the scandal of deduction, *Information* 4(1):33–59, 2013.

[10] D'Agostino, M., An informational view of classical logic, *Theoretical Computer Science* 606: 79–97, 2015.

[11] D'Agostino, M., M. Finger, and D.M. Gabbay, Semantics and proof-theory of depth bounded boolean logics, *Theoretical Computer Science* 480:43–68, 2013.

[12] D'Agostino, M., and L. Floridi, The enduring scandal of deduction, *Synthese* 167(2):271–315, 2009.

[13] D'Agostino, M., D.M. Gabbay, and S. Modgil, Normality, non-contamination and logical depth in classical natural deduction, *Studia Logica* 108(2):291–357, 2020.

[14] D'Agostino, M., and S. Modgil, Classical logic, argument and dialectic, *Artificial Intelligence* 262: 15–51, 2018.

[15] Dalal, M., Anytime clausal reasoning, *Annals of Mathematics and Artificial Intelligence* 22(3-4):297–318, 1998.

[16] Davey, B.A., and H.A. Priestley, *Introduction to Lattices and Order*, 2nd edition, Cambridge University Press, 2002.

[17] de Souza Rabello, G., and M. Finger, Approximations of modal logic K, *Electronic Notes in Theoretical Computer Science* 143:171–184, 2006.

[18] Finger, M., and D.M. Gabbay, Cut and pay, *Journal of Logic, Language and Information* 15(3):195–218, 2006.

[19] Finger, M., and R. Wassermann, The universe of propositional approximations, *Theoretical Computer Science* 355(2):153–166, 2006.

[20] Fischer, M.J., and R.E. Ladner, Propositional dynamic logic of regular programs, *Journal of Computer and System Sciences* 18(2):194–211, 1979.

[21] Garson, J.W., *Modal Logic for Philosophers*, Cambridge University Press, Cambridge, 2006.

[22] Ghidini, Ch., and L. Serafini, Distributed first order logic, *Artificial Intelligence* 253:1–39, 2017.

[23] Goubault, J., The complexity of resource-bounded first-order classical logic, in P. Enjalbert, E.W. Mayr, and K.W. Wagner, (eds.), *STACS 94, 11th Annual Symposium on Theoretical Aspects of Computer Science*, vol. 775 of LNCS, Springer, 1994, pp. 59–70.

[24] Grätzer, G., *Lattice Theory: Foundation*, Birkhäuser Basel, 2011.

[25] Grossi, D., and F.R. Velázquez-Quesada, Twelve Angry Men: A study on the finegrain of announcements, in X. He, J. Horty, and E. Pacuit, (eds.), *Proceedings of the 2nd international conference on logic, rationality and interaction*, vol. 5834 of LNCS, Springer, 2009, pp. 147–160.

[26] Hakli, R., and S. Negri, Proof theory for distributed knowledge, in F. Sadri, and K. Satoh, (eds.), *Computational Logic in Multi-Agent Systems, CLIMA VIII*, vol. 5056 of LNCS, Springer, 2007, pp. 100–116.

[27] Hammarberg, J., and S. Nadjm-Tehrani, Formal verification of fault tolerance in safetycritical reconfigurable modules, *International Journal on Software Tools for Technology Transfer volume* 7(3):268–279, 2005.

[28] Hawke, P., A. Özgün, and F. Berto, The fundamental problem of logical omniscience, *Journal of Philosophical Logic* 49(4):727–766, 2020.

[29] Indrzejczak, A., *Natural Deduction, Hybrid Systems and Modal Logics*, Springer, 2010.

[30] Jago, M., Epistemic logic for rule-based agents, *Journal of Logic, Language and Information* 18:131–158, 2009.

[31] Klassen, T.Q., S.A. McIlraith, and H.J. Levesque, Towards tractable inference for resource-bounded agents, in *2015 AAAI Spring Symposia, Stanford University, Palo Alto, CA, USA*, AAAI Press, 2015.

[32] Kleene, S.C., *Introduction to Metamathematics*, North-Holland, Amsterdam, 1952.

[33] Larese, C., *The Principle of Analyticity of Logic A Philosophical and Formal Perspective*, M.S. thesis, Scuola Normale Superiore, Pisa, Italy, 2019.

[34] Marques-Silva, J., and S. Malik, Propositional SAT solving, in E.M. Clarke, T.A. Henzinger, H. Veith, and R. Bloem, (eds.), *Handbook of Model Checking*, Springer, 2018, pp. 247–275.

[35] Marx, M., Complexity of modal logic, in P. Blackburn, J. van Benthem, and F. Wolter, (eds.), *Handbook of Modal Logic*, vol. 3 *of Studies in logic and practical reasoning*, North-Holland, 2007, pp. 139–179.

[36] Olkhovikov, G.K., and H. Wansing, Inference as doxastic agency. Part I: The basics of justification stit logic, *Studia Logica* 107(1):167–194, 2019.

[37] Prawitz, D., *Natural Deduction. A Proof-Theoretical Study*, Almqvist & Wilksell, Uppsala, 1965.

[38] Quine, W.V.O., *The Roots of Reference*, Open Court, 1973.

[39] Schaerf, M., and M. Cadoli, Tractable reasoning via approximation, *Artificial Intelligence* 74(2):249–310, 1995.

[40] Sheeran, M., and G. Stålmarck, A tutorial on Stålmarck's proof procedure for propositional logic, *Formal Methods in System Design* 16(1):23–58, 2000.

[41] Sipser, M., *Introduction to the theory of computation*, PWS Publishing Company, 1997.

[42] Solaki, A., Rule-based reasoners in epistemic logic, in J. Sikos, and E. Pacuit, (eds.), *At the Intersection of Language, Logic, and Information - ESSLLI 2018 Student Session*, vol. 11667 of LNCS, Springer, 2018, pp. 144–156.

[43] van Ditmarsch, H., and T. French, Semantics for knowledge and change of awareness, *Journal of Logic, Language, and Information* 23(2):169–195, 2014.

[44] van Ditmarsch, H., W. van der Hoek, and B. Kooi, *Dynamic Epistemic Logic*, vol. 337 of *Synthese Library*, Springer, 2008.

[45] Velázquez-Quesada, F., Dynamic epistemic logic for implicit and explicit beliefs, *Journal of Logic, Language and Information* 23(2):107–140, 2014.

[46] Whitsey, M., Logical omniscience: A survey, Technical Report NOTTCS-WP-2003-2, School of Computer Science and IT, University of Nottingham, 2003.

P. Pardo
University of Milan, Universita Degli Studi di Milano
Milano
Italy
`pere.pardo.v@gmail.com`