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 \( \text{NP} \)-complete resp. \( \text{coNP} \)-complete [8] and so under the widely accepted conjecture that \( \text{P} \neq \text{NP} \), 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 \( \text{DBBL} \) are tractable [10]. In the present paper, we focus on the informational approach to classical

\(^1\) Classical and epistemic logics suffer from \textit{logical omniscience} problems as models of reasoning agents [46]. Epistemic logic [44] is thus said to model what an agent \textit{can} know, or does \textit{implicitly} know, by purely deductive methods.
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 succinct form requiring less memory use and messaging time.

Given a proof \(\Pi\), say witnessing that \(\Delta \vdash B\), a deductive exchange format can be anything between: the full proof \(\Pi\) 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 succinctness, 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.
A Modal View on Resource-Bounded Propositional Logics

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 $\models \mathbb{S}$ that validate modus ponens for increasing sets $S$ of atoms; its valuations can be paraconsistent for $p \notin S$, i.e. $V(p) = 1 = V(\neg p)$.

In the same vein, Finger and Wassermann [19] describe a general system that validates each rule $\rho$ within a set of applicable formulas $S_\rho$ only; these sets $S_\rho$ (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 $\models_{\mathbb{S}}$ 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 $|=S$. 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.
A Modal View on Resource-Bounded Propositional Logics

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

$\begin{array}{c|ccc} & 1 & 0 & \bot \\ \hline 1 & 1 & 0 & \bot \\ 0 & 0 & 0 & 0 \\ \bot & \bot & 0, \bot & \bot \end{array}$

Figure 3. Truth-tables $f_*$ for $* \in \{\land, \lor, \neg, \to\}$. 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 $\text{Var} = \{p_1, \ldots, p_i, \ldots\}$, we define the language $\mathcal{L}$ as follows:

$\mathcal{L} := p_i \neg A | A \land B | A \lor B | A \rightarrow B$

We also use the logical constant $\bot$ (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. $\bot \lor \bot \in \{1, \bot\} = f_\lor(\bot, \bot)$. Under this reading, $A$ and $B$ can be unknown ($A, B \mapsto \bot$) but their disjunction known $A \lor B \mapsto 1$ (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 $\{A \rightarrow B, \neg A \lor B\}$. This is not the case if $\mathcal{L}$ is defined from minimally functionally complete sets, e.g. $\{\neg, \lor\}$ defining $A \rightarrow B := \neg A \lor B$ and so on.

$^6$ The truth-tables in Kleene’s 3-valued logic are as in Figure 3 except for: $\bot \lor \bot = \bot \land \bot = \bot \rightarrow \bot = \bot$. Quine [38] motivates the informational reading of $\lor$ and $\land$ 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 \( V : \mathcal{L} \to \{ \top, \bot \} \) 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 \( \{ \bot \} \), from which the 0-depth logic \( (\mathcal{L}, \vdash_0) \) is defined as usual.

Definition 1.3. (\( \vdash_0 \) consequence) Given \( \Gamma \cup \{ A \} \subseteq \mathcal{L} \), we say that \( A \) is a 0-depth consequence of \( \Gamma \), denoted \( \Gamma \vdash_0 A \), if for any valuation \( V \), \( V[\Gamma] = \{ \top \} \) implies \( V(A) = \top \).

\( \Gamma \) is 0-depth consistent if there is a valuation \( V \) with \( V[\Gamma] = \top \). (Note we make use of the notation \( f[X] = \{ f(x) : x \in X \} \).)

The logic \( \vdash_0 \) is a Tarskian consequence relation, as it satisfies reflexivity, monotonicity and cut [11]. \( \vdash_0 \) is also structural but has no tautologies. Classical inconsistencies are assigned \( \bot \) by \( V_\bot \), \( \bot \) by any two-valued valuation \( V : \mathcal{L} \to \{ \top, \bot \} \), and even \( \top \) if 0-depth consistent. Still, 0-depth inconsistent sets are explosive: \( \{ p, \neg p \} \vdash_0 \bot \).

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

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

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

\[
\begin{array}{c}
A_1 \\
A_0 \\
A_2
\end{array}
\]

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.
A Modal View on Resource-Bounded Propositional Logics

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 \( * \in \{\land, \lor, \rightarrow, \neg, \land\} \). 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

\[
\begin{array}{cccc}
\frac{A \quad B}{A \land B} & \frac{\neg A}{\neg (A \land B)} & \frac{\neg B}{\neg (A \land B)} & \frac{A}{\neg \neg A} \\
\frac{\neg A \quad \neg B}{\neg (A \lor B)} & \frac{A}{A \lor B} & \frac{B}{A \lor B} & \frac{A \quad \neg A}{\land} \\
\frac{A \quad \neg B}{\neg (A \rightarrow B)} & \frac{\neg A}{A \rightarrow B} & \frac{B}{A \rightarrow B} & \frac{A \rightarrow B}{A \rightarrow B}
\end{array}
\]

elimination:

\[
\begin{array}{cccc}
\frac{\neg (A \land B) \quad A}{\neg A} & \frac{\neg (A \land B) \quad B}{A \land B} & \frac{A \land B}{A} & \frac{A \land B}{B} & \frac{\neg A}{A} \\
\frac{A \lor B \quad \neg A}{\neg B} & \frac{A \lor B \quad \neg B}{A} & \frac{\neg (A \lor B)}{A} & \frac{\neg (A \lor B)}{B} & \frac{A \quad \land}{A}
\end{array}
\]

\[
\begin{array}{cccc}
\frac{A \rightarrow B \quad A}{B} & \frac{A \rightarrow B \quad \neg B}{\neg (A \rightarrow B)} & \frac{\neg (A \rightarrow B)}{A} & \frac{\neg (A \rightarrow B)}{B}
\end{array}
\]

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

\[
\Gamma \models_k B \quad \text{iff} \quad \text{there is } A \in \mathcal{L} \text{ such that } \Gamma \cup \{A\} \models_k B \text{ and } \Gamma \cup \{\neg A\} \models_k B.
\]

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 \( \Gamma \) is a tree \( (N, R) \) as in Definition 1.4 satisfying conditions (i)–(ii) with rules from Figures 4 and 5, and also:

\[
\begin{array}{cccc}
\{A\}^1 & \{\neg A\}^2 & \Gamma, \{A\}^1 & \Delta, \{\neg A\}^2 \\
\vdots & \vdots & \vdots & \vdots \\
B & B & B & B
\end{array}
\]

\[
\begin{array}{cccc}
\vdots & \Pi_1 & \vdots & \Pi_2 \\
B & B & B & B
\end{array}
\]
(iii) each branch of the tree traverses \( \leq k \) applications of \( RB \).

Example 2. (\( RB \) and depth) Since \( \{p\} \vdash_0 p \lor \neg p \) and \( \{-p\} \vdash_0 p \lor \neg p \), an application of \( RB \) with the (discharged) pair \( \{p, \neg p\} \) gives \( \emptyset \vdash_1 p \lor \neg p \). 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 \( \text{sub}(\Gamma \cup \{B\}) \) or atoms \( \text{at}(\Gamma \cup \{B\}) \).

Definition 1.8. (Virtual space function; \( \models^f_k \)) A virtual space function is a function \( f : \mathcal{P}_{\text{fin}}(\mathcal{L}) \rightarrow \mathcal{P}(\mathcal{L}) \) satisfying: (i) \( \text{at}(\Delta) \subseteq \text{f}(\Delta) \), (ii) \( \text{f}(\Delta) \) is closed under subformulas, and (iii) the size of \( \text{f}(\Delta) \) is polynomially bounded w.r.t. \( \Delta \). Any such function \( f \) defines a consequence \( \models^f_k \) and calculus \( \vdash^f_k \) as follows. Let \( \models^0_0 = \models_0 \) resp. \( \vdash^0_0 = \vdash_0 \) and:

\[
\Gamma \models^f_{k+1} B \text{ iff there is } A \in \text{f}(\Gamma \cup \{B\}) \text{ with } \Gamma \cup \{A\} \models^f_k B \text{ and } \Gamma \cup \{-A\} \models^f_k B;
\]

\[
\Gamma \vdash^f_{k+1} B \text{ iff } \Gamma \vdash_{k+1} B \text{ holds with } RB\text{-formulas satisfying } A \in \text{f}(\Gamma \cup \{B\}) \mbox{.}\]

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 \( |\Gamma| = \sum_{A \in \Gamma} |A| \).

Proposition 1.9. (Approximation; [10, Prop. 3.2]) For any virtual space function \( f \), it holds that \( \models = \bigcup_k \models^f_k \).

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 \( \Gamma \vdash^f_k A \) is \( O(n^{k+2}) \).

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

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.
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 $\vdash_k$-proof of $B$ obtains from applying $RB$ to all pairs $\{p_i, \neg p_i\}$ occurring in $\Gamma$, and the interim rules $\{E\to, I\wedge\}$.

| $\Gamma$ | $\Gamma \models_k B$ | $h(\Gamma, B)$ |
|----------|-------------------|---------------|
| $\{\pm p_1 \to q, \quad q \wedge \pm p_2 \to B\}$ | $2 \leq k$ | $2$ |
| $\{\pm p_1 \to q_1, \pm p_2 \to q_2, \quad q_1 \wedge p_3 \to B, \quad q_2 \wedge \neg p_3 \to B\}$ | $2 \leq k$ | $3$ |
| $\{\pm p_1 \to q_1, q_1 \wedge \pm p_2 \to q_2, \quad q_2 \wedge \pm p_3 \to B\}$ | $3 \leq k$ | $3$ |

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

$$h(\Gamma, B) = \text{the minimum number of formula pairs } \{A, \neg A\} \text{ needed as }$$

$RB$-assumptions to show that $\Gamma \models B$.

**Remark 1.** (Notation) ($\pm$ 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 $\pm A_1, \ldots, \pm A_k$.

($\pm$ inside formulas.) In examples, $\pm$ is used for succinctness: 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 $(\Gamma, B)$, denoted $h(\Gamma, B)$, is the minimum $n \in \mathbb{N}$ such that there exist $A_1, \ldots, A_n$ with $\Gamma \cup \{\pm A_1, \ldots, \pm A_n\} \models B$.

**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:

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

**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 $k$. ($\Rightarrow 1$) 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 $\Gamma \cup \{-A_{k+1}\} \models_k B$, so Definition 1.6 gives $\Gamma \models_{k+1} B$.

$(\Rightarrow)$ Now assume $\Gamma \models_{k+1} B$, so $\Gamma \cup \{A\} \models_k B$ and $\Gamma \cup \{-A\} \models_k B$, for some $A$. Applying twice the IH, we obtain two sets of formulas such that

$$ (\Gamma \cup \{A\}) \cup \{\pm A_1, \ldots, \pm A_{2^{k+1} - 1}\} \models 0 B \quad \text{and} \quad (\Gamma \cup \{-A\}) \cup \{\pm A_1', \ldots, \pm A'_{2^{k+1} - 1}\} \models 0 B. $$

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, $\Gamma \cup \{\pm A_1, \ldots, \pm A_{2(2^{k-1}) + 1}\} \models 0 B$. Since $2 \cdot (2^k - 1) + 1 = 2^{k+1} - 2 + 1 = 2^{k+1} - 1$ we are done. \[\blacksquare\]

A hierarchy of degree-bounded logics $\langle \models_k \rangle_{k, \omega}$ satisfying $\Gamma \models_k B$ 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 $\mathcal{L}$: $\mathbb{V} = \{V : \mathcal{L} \to \{1, 0, \bot\} \mid V \text{ is a valuation}\}$.

**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 $V(A) = \bot$ or $V'(A) = V(A)$.

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

**Example 4. (Meet and join)** For the poset $(\{1, 0, \bot\}, \leq)$, its meets are: $\sqcap\{v\} = v$ and otherwise $\sqcap X = \bot$; joins are $\sqcup\{v\} = v = \sqcup\{v, \bot\}$ while $\sqcup\{1, 0\}$, $\sqcup\{1, 0, \bot\}$ 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 $\sqcap\mathbb{V}' = \{\langle A, \sqcap_{V \in \mathbb{V}'} V(A) \rangle : A \in \mathcal{L}\}$. 

P. Pardo
A Modal View on Resource-Bounded Propositional Logics

Figure 6. The information order $\sqsubseteq$ on the set of 3-valued functions $f : \mathcal{L} \rightarrow \{1, 0, \bot\}$. Its maximal elements are the 2-valued functions $f : \mathcal{L} \rightarrow \{1, 0\}$, including the constant function $V_\lambda : \mathcal{L} \rightarrow \{1\}$ and the set $\mathcal{C}$ of classical valuations. The $\sqsubseteq$-minimum element is $V_\bot : \mathcal{L} \rightarrow \{\bot\}$. ($\mathcal{V}, \sqsubseteq$) 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 $\bigsqcup Y = \bigsqcup Y^u$. (That is, the meet of the set $Y^u$ of upper bounds of $Y$, Definition A.2.)

The set $\mathcal{V}$ is closed under meets (Proposition 3.4) but lacks a top element that acts as the join $\mathcal{V} \cup V'$ 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, $V_\lambda$, similarly makes all formulas true.

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

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

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

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

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

10Thus, the truth-value of $B$ in $\bigsqcap (\mathcal{V}' \cup \{V_\lambda\})$ is no longer its pointwise meet, but the pointwise meet of $\bigsqcap \mathcal{V}'$. For an illustration, the pointwise meet of $\{V_\lambda, V_{\lnot p}\}$ would give $V_\lambda(p) \cap V_{\lnot p}(p) = 1 \cap 0 = \bot$ instead of $(V_\lambda \cap V_{\lnot p})(p) = V_{\lnot p}(p) = 0$.
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 $\subseteq$ and $\subseteq$.

**Definition 4.1. (Theory)** The set of 0-depth consequences of $\Gamma$ is $Cn_0(\Gamma) = \{ A \in \mathcal{L} : \Gamma \vdash_0 A \}$. A set $\Gamma \subseteq \mathcal{L}$ is a theory if $Cn_0(\Gamma) \subseteq \Gamma$. Abusively, we define the theory of a valuation $V$ as the set $Th(V) = \{ A \in \mathcal{L} : V(A) = 1 \}$.

**Fact 4.2.** For any $V \in \mathcal{V}^+$, $Th(V)$ is a theory: $Cn_0(Th(V)) \subseteq Th(V)$.

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

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

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

Whenever $\Gamma$ is 0-depth consistent, $\mathcal{V}_\Gamma$ will contain some valuation in $\mathcal{V}$. All 0-depth inconsistent sets $\Gamma$ share the same set of valuations $\mathcal{V}_\Gamma = \{ V_\lambda \}$. In both cases, the infimum $\bigcap \mathcal{V}_\Gamma$ (Proposition 3.4) happens to be in $\mathcal{V}_\Gamma$ and so it is a minimum: $V_\Gamma = \bigcap \mathcal{V}_\Gamma$. 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 $\subseteq$-minimum valuation, called $V_\Gamma$, exists in $\mathcal{V}_\Gamma$. As a consequence $V_\Gamma = \bigcap \mathcal{V}_\Gamma$.

**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 $V_\Gamma$ suffices for checking any $\models_0$-consequence $A$ of $\Gamma$, 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) $\mathcal{V}_\Gamma = \uparrow V_\Gamma$ and (ii) $\bigcap \uparrow V_\Gamma = V_\Gamma$.

**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 $\Gamma$, it holds that $\Gamma = Th(V_\Gamma)$. (ii) For any $V' \in \mathcal{V}$, it holds that $V' = V_{Th(V')}$. 

**Proof.** We use that $\Gamma$ 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 \( A \in \text{Th}(V_\Gamma) \);
(ii) \( V'(A) = 1 \) iff \( A \in \text{Th}(V') \) iff \( \text{Th}(V') \models_0 A \) iff \( V_{\text{Th}(V')}(A) = 1 \)

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

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

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

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

The connectives \( \land, \lor \) need not be distributive over each other, as witnessed by valuations \( V \) satisfying one of the following:

\[
V(\bot \land (1 \lor 1)) = \bot < 1 = V((\bot \land 1) \lor (\bot \lor 1)) \in \{1, \bot\}
\]

\[
\{\bot, 0\} \ni V(0 \lor (\bot \lor \bot)) = \bot < 0 = V((0 \lor \bot) \land (0 \lor \bot)) \in \{0, \bot\}.
\]

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) \( \bigsqcup V' \) is either \( V_\wedge \) or the pointwise join \( \{\langle A, \bigsqcup_{V \in V'} V(A) \rangle : A \in \mathcal{L} \} \).

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

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

\[
(\bigsqcup V')(A) = 1 \text{ iff } A \in \bigcup_{V \in V'} \text{Th}(V) \text{ iff } (\bigcup_{V \in V'} V(A)) = 1.
\]

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

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 \( \uparrow V \cap V_{\{A\}} \). 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 \( V, V_{\{A\}} \subseteq V \sqcup V_{\{A\}}, V \sqcup V_{\{A\}} \) 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 \( V^A = V_\bot \).

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

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

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

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

\[
V_{\Gamma} \sqsubseteq V_{\Gamma \uparrow \neg p} \subseteq V_{\Gamma}^{[\pm]p} \subseteq V_{\Gamma}^{p \neg p} \sqsubseteq V_{\Gamma}^{p \neg p}
\]

\[
\Gamma \quad p \lor \neg p \quad q \quad p \neg p.
\]

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 \( \Gamma_1 = \{\pm p \rightarrow q, p \lor \neg p\} \) and resp. \( \Gamma_3 = \{\pm p \rightarrow q, p\} \) give:

\[
V_{\Gamma_1} = V_{\Gamma_1}^{p \lor \neg p} \quad \text{and} \quad V_{\Gamma_3}^{[\pm]p} = V_{\Gamma_3}^{p \lor p} (= V_{\Gamma_3}).
\]

Example 6. ((\( \forall \Gamma, \sqsubseteq \) is not distributive) The inequality \( V_{\Gamma}^{p \lor \neg p} \sqsubseteq V_{\Gamma}^{[\pm]p} \) seen above for \( \Gamma = \{\pm p \rightarrow q\} \) shows that the lattice of valuations is not
Lemma 5.8. \( V \cap V_{\neg p} = V \cup (V_{\neg p}) \cap (V_{\neg p}) \) does not generalize into \( V_{A \lor B} = V_{\land} \land V_{\lor} \) e.g. for \( A = p = B \). The general property only holds if 0-depth logic is expanded with \textit{idempotence} \( A \lor A \rightarrow A \), an option discussed in [12].
Corollary 5.9. (Permutation of ±-updates) For any valuation $V \in \mathcal{V}$ 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 ⊑ and ⊒. For (⊒),

$(V^{[\pm]A})^{[\pm]B} \sqcap (V^{[\pm]A})^{\neg B} = (V^{[\pm]A} \sqcup V_{\{B\}}) \sqcap (V^{[\pm]A} \sqcup V_{\{\neg B\}})$

$\equiv (V^{A} \sqcap V^{\neg A}) \sqcup V_{\{B\}} \sqcap (V^{A} \sqcup V_{\{\neg B\}}) \sqcap (V^{\neg A} \sqcup V_{\{\neg B\}})$

$= (V^{A}B) \sqcap (V^{\neg A}B) \sqcap (V^{A}B) \sqcap (V^{\neg A}B)$

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

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

6. **UBBL: A Hierarchy of Logics of Bounded Degree**

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

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

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

Definition 6.2. ($\vdash^{f}_{k}$-proof) Given a virtual space function $f$, a $\vdash^{f}_{k}$-proof of $B$ from $\Gamma$ is a tree $(N, R)$ satisfying (i)–(ii) from Definition 1.7 plus:

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

Lemma 6.3. (Soundness, completeness UBBL) For any $k$, any function $f$ and set $\Gamma \cup \{B\} \subseteq \mathcal{L}$, it holds that $\Gamma \models^{f}_{k} B$ iff $\Gamma \models^{f} B$. 
\textbf{Proof.} (Base case 0) From definitions and Proposition 1.5, $\models f_0 = \dashv 0 = \models f_0$. (Ind. case $k+1$.) Assume the claim holds for $k$ and let $\Gamma \vdash f_{k+1} A$. Given a $\models f_{k+1}$-proof of $B$ from $\Gamma$, one can split all $RB$-instances with, say, $\{A_1, \neg A_1\}$ and obtain two proofs witnessing $\Gamma \cup \{A_1\} \models_k B$ resp. $\Gamma \cup \{-A_1\} \models_k B$ whose $RB$-instances are all within a set $\{\pm A_2, \ldots, \pm A_{k+1}\} \subseteq f(\Gamma, B)$. 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 $A_1, A_2, \ldots A_{k+1} \in f(\Gamma \cup \{B\})$:

$$\Gamma \models f_{k+1} B$$

iff $\Gamma \cup \{A_{k+1}\} \models f_k B$ and $\Gamma \cup \{-A_{k+1}\} \models f_k B$

iff $V[\pm A_1 \ldots [\pm A_k](B) = 1$ for all $V \in \bigwedge \cup \{A_{k+1}\} \cup \bigwedge \cup \{-A_{k+1}\}$

iff $V[\pm A_1 \ldots [\pm A_k](B) = 1$ for all $V \in (\bigwedge) A_{k+1} \cup (\bigwedge) -A_{k+1}$

iff $(V[\pm A_1 \ldots [\pm A_k]) A_{k+1}(B) = 1$ for all $V \in \bigwedge$ and

$(V[\pm A_1 \ldots [\pm A_k]) -A_{k+1}(B) = 1$ for all $V \in \bigwedge$

iff $(V[\pm A_1 \ldots [\pm A_k]) (\pm A_{k+1})(B) = 1$ for all $V \in \bigwedge$

iff $\Gamma \models f_{k+1} B$\

where each step follows from: the reasoning above; the inductive hypothesis; Lemma 5.4; def. of $(\bigwedge) A$; Definition 5.5 and meets; and resp. Definition 6.1.

\[\mathbf{F a c t 6.4.}\] It holds that $\models_1 = \models_1$ and $\models_k \varsubsetneq \models_k$ for any $k > 1$.

\textbf{Proposition 6.5.} (Approximation) $\models = \bigcup_k \models f_k$, for any $f$ as in Definition 1.8.

\textbf{Proposition 6.6.} (Bounded transitivity) For any $\Gamma \cup \{A, B\} \subseteq \mathcal{L}$ with $A \in f(\Gamma \cup \{B\})$, if $\Gamma \models f_i A$ and $\Gamma \cup \{A\} \models f_j B$ then $\Gamma \models f_{i+j} B$.\[12\]

\textbf{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 $\Gamma \models f_k A$ is $O(n^{k+2})$.

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

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

\[\mathbf{Note}\] that the corresponding claim in [10] is missing the assumption $A \in f(\Gamma \cup \{B\})$.\[12\]
Lemma 6.9. For any $V \in \mathcal{V}$, the following are equivalent:

(i) $V \not\sqsubseteq \overline{V}$, for any $\overline{V} \in \mathcal{C}$;
(ii) $V^{\lbrack \pm A_1 \cdots \pm A_k \rbrack} = V_\wedge$, 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 $\mathcal{V}'$ its join is the top element $\bigsqcup \mathcal{V}' = V_\wedge$ if $\mathcal{V}'$ contains either: $V_\wedge$, 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 $V_1(A_1) \neq \perp, \ldots, V_k(A_k) \neq \perp$.

Example 8. (Inconsistency) The disjunction $(p_1 \land \neg p_1) \lor \ldots \lor (p_{k+2} \land \neg p_{k+2})$ 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 $p_{k+2} \land \neg p_{k+2}$ true, thus making $V^{\lbrack \pm p_1 \cdots \pm p_{k+1} \rbrack} = V_\wedge$.

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 $C \in \mathcal{L}$:

$$\mathcal{L}_U := p_i \mid \neg A \mid A \land B \mid A \lor B \mid A \rightarrow B \mid [\pm C]A$$
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 $V : \mathcal{L}_U \rightarrow \{1, 0, \bot\}$ 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 $V' \supseteq V$ and $V'(A) = 1$. We also call the function $V_\lambda : \mathcal{L}_U \rightarrow \{1\}$ 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:

$$\Gamma \models_{\mathbb{U}} A \text{ iff } V[\Gamma] = \{1\} \text{ implies } V(A) = 1 \text{ for any } V \in \mathbb{U}$$

$$\Gamma \models_{\mathbb{U}} A \text{ iff } \forall \Gamma \models_{\mathbb{U}} A \text{ for any } U$$

**Example 9.** (Valuations) All $\mathcal{L}_U$-valuations $V$ satisfy $V([\pm p]p \lor \neg p) = V^[\pm p](p \lor \neg p) = 1$. Thus, their minimum $V_0$ (see Corollary 8.9) is not the null function $f_\bot : \mathcal{L}_U \rightarrow \{\bot\}$ (Figure 7), although $V_0$ does contain $V_\bot : \mathcal{L} \rightarrow \{\bot\}$.

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

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

**Fact 7.4.** (Trivial and generated $U$-models) The set $\mathbb{U}_\lambda = \{V_\lambda\}$ is a $U$-model with $\models_{\mathbb{U}_\lambda} = \mathcal{P}(\mathcal{L}_U) \times \mathcal{L}_U$. 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 $\mathcal{L}_{U_k}$ of the language $\mathcal{L}_U = \bigcup_{k<\omega} \mathcal{L}_{U_k}$.

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

$$\mathcal{L}_{Uk+1} := [\pm A]B_i \quad \neg B \quad B \land C \quad B \lor C \quad B \rightarrow C.$$

where $A \in \mathcal{L}$ and $B_i \in \mathcal{L}_{U \leq k} := \bigcup_{i \leq k} \mathcal{L}_{Ui}$. 
Fact 7.7. The standard model $S$ is closed under updates and meets, and so is a $U$-model. Moreover, $S$ is the set of all valuations. Hence, $\models_U = \models_S$.

Proof. Observe first that $S_n$ is a set of functions $V_n : L_{U \leq n} \to \{1, 0, \bot\}$. 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 $L_{U \leq n}$, together with the auxiliary claim that $S_n$ is closed under meets and updates.

(Base case $n = 0$.) Since $S_0 = \emptyset$ and $L_{U0} = L$, $S_0$ is a set of $L_{U0}$-valuations, and it is closed under updates (Proposition 5.2) and meets (Proposition 3.7).

Assume as inductive hypothesis that the claims hold for $S_n$.

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

$$\bigcap S'_{n+1}([\pm B]C) = \bigcap_{V_{n+1} \in S'_{n+1}} V_{n+1}([\pm B]C) = \bigcap_{V \in S'_{n}} (V^B \cap V^\neg B)(C) = (\bigcap_{V \in S'_{n}} V^B \cap \bigcap_{V \in S'_{n}} V^\neg B)(C).$$

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

Let now $V = \bigcup \tilde{V}$ for some $\subseteq$-chain $\tilde{V} = \langle V_n \rangle_{n \in \omega}$ with $V_n \in S_n$. Since each finite fragment $V_n$ satisfies the truth-tables for $L_{U \leq n}$-formulas, so does
V for $\mathcal{L}_U$-formulas. The function $V$ also satisfies the update condition: say $C \in \mathcal{L}_{U_0}$; then $V([\pm B]C) = V_{n+1}([\pm B]C) = (V_n^B \cap V_{n-1}^B)(C)$. This shows that $\mathcal{S}$ is a set of valuations. ($\mathcal{S}$ is the set of all valuations.) Vice versa, for any valuation $V \neq V_\lambda$, $V$ satisfies the truth-tables and the update condition over $\mathcal{L}_U$ and so over every finite language $\mathcal{L}_{U_n}$; hence each finite fragment $V_n = V \cap (\mathcal{L}_{U_n} \times \{1, 0, \perp\})$ 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 $V_n \in \mathcal{S}_n$.

**Corollary 7.8.** (Complete lattice) $(\mathcal{S}, \sqsubseteq^+)$ is a complete lattice, where $\sqsubseteq^+ = \sqsubseteq \cup (\mathcal{S} \times \{V_\lambda\})$. Hence, for any $\mathcal{S}' \subseteq \mathcal{S}$ its join is $\bigcup \mathcal{S}' = \bigcap (\mathcal{S}')^\uparrow$.

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

**Proof.** The case $V = V_\lambda$ is trivial. For any other valuation: $V(\neg[\pm A]\neg B) = 1 \iff V([\pm A]\neg B) = 0 \iff (V^A \cap V^{-A})(\neg B) = 0 \iff V^A(\neg B) = 0 \iff V^{-A}(\neg B) \iff V^A(B) = 1 = V^{-A}(B) \iff (V^A \cap V^{-A})(B) = 1 \iff V([\pm A]B) = 1 \iff V([\pm A]B) = 1$. 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 $V(\neg[\pm A]\neg B) = \perp$ iff $V([\pm A]B) = \perp$. In either case, $V(\neg[\pm A]\neg B) = V([\pm A]B)$.

**Definition 7.10.** ($\vdash_U$ system) A natural deduction system for $\vdash_U$ 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 $\vdash_U$ consequence.

**Proof.** For an arbitrary $U$-model $\mathcal{U}$, 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 $\Gamma \cup \{A\} \models_U B$ and $\Gamma \cup \{\neg A\} \not\models_U B$, and let $V \in \mathcal{U}_\Gamma$ be arbitrary. By Definition 7.2, $V^A \in \mathcal{U}$ is the $\sqsubseteq$-minimum element with (i) $V^A \sqsubseteq V$ and (ii) $V^A(A) = 1$. From (i), $V^A[\Gamma] = \{1\}$, and so with (ii)
we obtain $V^A[\Gamma \cup \{A\}] = \{1\}$. This implies that $V^A \in \mathbb{V}_{\Gamma \cup \{A\}}$ and using the initial assumption $V^A(B) = 1$. With a similar reasoning for $V^{\neg A}$, we conclude $V^A(B) \cap V^{\neg A}(B) = 1$ and so $V([\pm A]B) = 1$.

$(E[\pm])$ Let $V \in \mathbb{V}$ 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 $V^{\neg A} = V_\lambda$ 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.

$(Red\neg, Red\neg')$ 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 $\{B, C\} \vdash A$ 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 $\models_U$. 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 $A \vdash_B B$ and $B \vdash A$.

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

$$
\begin{align*}
[\pm A]B & \quad O(\pm A) \\
\vdots & \\
[\pm A]C & \quad C(\pm A) \\
\hline
[\pm D]A & \quad B \\
& \quad \begin{cases}
[\pm I/E*] & \quad [\pm D]C \\
[\pm E] & \quad [\pm A]B
\end{cases}
\end{align*}
$$

\begin{align*}
\frac{[\pm A]B \lor B}{\neg[\pm D]A} & \quad E\lor \\
\frac{[\pm D]A \lor B}{\neg[\pm D]A} & \quad \text{Contr.}
\end{align*}

Figure 9. A list of rules derivable in $\models_U$. $O(\pm A), C(\pm A)$ open and close a subproof; any formula above the subproof can be imported into it. $D, Alt, Triv$ 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\} $\vdash A$ with modal premises: $[\pm D]B$ or $[\pm D]C$ or both. $Contraction$ and $Collapse$ show that any two non-empty sequences of modalities $[\pm A], [\pm A]$, indistinctly written $([\pm A]*([\pm A]*)*)$, 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 $A \equiv_0 \neg B$.
$E \lor$ treats the modal formula $[\pm D]A$ as a block, while $[\pm]E \lor$ ignores the prefix $[\pm D]$, applies $E \lor$, 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 $\Pi$ be a proof of $C$ from $\{B\} \cup \Delta$. Prefix $\Pi$ with an application of $E[\pm]$ with $A$ to obtain a proof $\Pi_1$ of $C$ from $\{A,[\pm A]B\} \cup \Delta$. A similar reasoning with $\neg A$ gives also a proof $\Pi_2$ of $C$ from $\{\neg A,[\pm A]B\} \cup \Delta$. Combine $\Pi_1, \Pi_2$ 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 $\Delta$ 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 $\neg[\pm A] \neg\neg B$, i.e. $[\pm A]B$.

**(Alt).** From $\neg[\pm A] \neg\neg B$, 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 $\Pi_{\neg A,B}, \Pi_{\neg A,\neg B}$ and $\Pi_{A,\neg B}$ of $C$ are obtained from the corresponding assumptions. Applying $I[\pm]$ to $\Pi_{A,B}, \Pi_{\neg A,B}$ and in parallel to $\Pi_{\neg A,\neg B}, \Pi_{A,\neg B}$ 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 $[\pm A][\pm B]C$ discharging $\{B, \neg B\}$.

**(Triv).** A proof of $B$ from $\Gamma$ 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 $\Gamma$.

**([\pm]I*) or ([\pm]E*).** The derivability of each modal intelim rule can be routinely checked. We only illustrate the case $[\pm]E \lor$ above. From $\{[\pm D](A \lor B), \neg B\}$, open a subproof with $A \lor B$, import $\neg B$ and apply $E \lor$ 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 $([\pm A]^*([\pm A]^*]^*B$ 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):
where \((\ast)\) turns \([\pm A]([\pm A]B)\) into two proofs of \([\pm A]B\) 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 \(\Gamma\).

\((RE)\). 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 \(\Pi_{AB}\) of \(A \vdash_0 B\) exists consisting solely of single premise rules. Since all these rules \(C \vdash D\) in Figure 4 admit contraposition \(\neg D \vdash \neg C\), reversing this proof gives a proof \(\Pi_{AB}^\top\) of \(\neg B \vdash_0 \neg A\). Let then \(\Pi\) be a proof of \([\pm A]C\). Combined with \(\Pi_{BA}\), an application of \(E[\pm]\) gives a proof \(\Pi_1\) of \(C\) from \(B\). Another combination with \(\Pi_{BA}^\top\) gives a proof \(\Pi_2\) of \(C\) from \(\neg B\). Finally, combine \(\Pi_1, \Pi_2\) 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 \(\Pi\) with each proof \(\Pi_{BA}^\top\) and \(\Pi_{BA}^{\top\top}\) 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\(^{14}\):

\[
\begin{align*}
\text{Axiom K fails:} & \quad \not\vdash_U [\pm p] (p \to p) \to ([\pm p]p \to [\pm p]p) \\
\text{Distribution fails:} & \quad \not\vdash_U ([\pm p] (p \lor \neg p) \to ([\pm p] p \lor \langle \pm \rangle \neg p).}
\end{align*}
\]

8. Completeness and Conservativeness of the \(\vdash_U\) Logic

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

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

Define the \(\bot\)-completion of a partial function \(f : \mathcal{L}_U \to \{1, 0, \bot\}\) as \(\bot(f) = f \cup \{(A, \bot) : A \not\in \text{dom}(f)\}\).

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

\[
\begin{align*}
V^0_\Delta &= (\mathcal{C}_U(\Delta) \times \{1\}) \cup \{(B, 0) : \neg B \in \mathcal{C}_U(\Delta)\} \\
V^{n+1}_\Delta &= (\mathcal{C}_U(I[\pm](\Delta, n)) \times \{1\}) \cup \{\{B : \neg B \in \mathcal{C}_U(I[\pm](\Delta, n))\} \times \{0\}\end{align*}
\]

\(^{13}\)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 \(A \lor A \vdash A\). A are added to \(\vdash_0\).

\(^{14}\)A version of axiom \(K\) as a rule \(\{[\pm A](B \to C), [\pm A]B\} \vdash [\pm A]C\) is valid, though, as a modal intelim rule \([\pm]E\).
where $I[\pm](\Delta, n) = \text{Th}(V^m_{\Delta}) \cup \bigcup_{A \in L} \{[\pm A]B : \langle B, 1 \rangle \in V^n_{\Delta \cup \{A\}} \cap V^n_{\Delta \cup \{-A\}}\}$. Finally, define: $V^\prime_{\Delta} = \bigcup_n V^n_{\Delta}$ and $V_{\Delta} = \bot(V^\prime_{\Delta})$.

**Definition 8.3. (Canonical model)** The canonical model is the set of all $\vdash_U$-closed functions $M = \{V_{\Gamma} : \Gamma \subseteq L_U\}$ built using Definition 8.2. Its $\subseteq$-minimum element is $V_0$. Its $\subseteq$-maximum element is $V_{\{p \land \neg p\}}$, abusively denoted $V_{\lambda}$.

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

**Proof.** ($\Rightarrow$). Let $\Pi$ be a proof of $A$ from $\Gamma$, and let $\Pi_1, \ldots, \Pi_m$ be all the subtrees of $\Pi$ with an instance of $I[\pm]$ at the root node, say for a conclusion $[\pm B_i]C_i$ in $\Pi_i$. 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 $\Pi_{i+1}$ 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 $\Pi$ that respects the inverse tree order: if $\{B, C\} : A$ occurs in $\Pi$, 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 $A_i^0$:

$$\langle A_1^0, \ldots, A_n^0 \rangle, A_0^0 = [B_1]C_1, A_1^1, \ldots, A_n^1, \ldots, A_0^m = [B_m]C_m, A_1^m, \ldots, A_n^m$$

From the original proof $\Pi$ 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 $A_j^i$ in the above sequence, $V^i_{\Gamma \cup \{\pm B_{i+1}, \ldots, \pm B_m\}}(A_j^i) = 1$ (where $V^i$ is as in Definition 8.2). The proof will conclude with the goal $V^m_{\Gamma \cup \{\pm B_{i+1}, \ldots, \pm B_m\}}(A) = 1$ if any $A = A_n^m$. For convenience let $\Gamma(i)$ denote any extension of $\Gamma$ 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 $A_1 = A_1^i \in \Gamma(1)$ 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 $\Gamma(2) \cup \{-B_1\} \vdash_U C_1$. Since $\Pi_1$ contains no other instance of $I[\pm]$, there are $\vdash_U$-proofs $\Pi_i', \Pi_{i''}$ of these two claims without $I[\pm]$. Thus, $V^0_{\Gamma(2) \cup \{B_1\}}(C_1) = 1 = V^0_{\Gamma(2) \cup \{-B_1\}}(C_1)$ and therefore $V^1_{\Gamma(2)}([\pm B_1]C_1) = 1$.

(Ind. case $A_{k+1}$.) Assume as inductive hypothesis that for each $A_k = A_j^i$ with $k' \leq k$ it holds that $V^i_{\Gamma(i+1)}(A_j^i) = 1$. Let now $A_k = A_j^i$. In case $A_{k+1} = A_j^{i+1}$, then $A_{k+1}$ follows in $\Pi$ from an intelim rule, $E[\pm]$, $\text{Red}_\neg$ or $\text{Red}_\neg'$ 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 the formula is derived in $\Pi$ 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 $\Gamma(i + 2)$ and $\{B_{i+1}\}$, and from $\Gamma(i + 2)$ and $\{\neg B_{i+1}\}$. 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 \( V_{\Gamma(i+1)}^i(C_{i+1}) = 1 \). In particular, \( V_{\Gamma(i+2)}^i(\{B_{i+1}\}(C_{i+1}) = 1 \) and \( V_{\Gamma(i+2)}^i(\{B_{i+1}\}(C_{i+1}) = 1 \). Thus, \( \langle C_{i+1}, 1 \rangle \in V_{\Gamma(i+2)}^i(\{B_{i+1}\}(C_{i+1}) \) and so by definition \( [\pm B_{i+1}]C_{i+1} \in I V \) \( (\Gamma(i+2), i) \) and finally \( V_{\Gamma(i+2)}^i(\{B_{i+1}\}(C_{i+1}) = 1 \).

(\( \Leftarrow \)). This is immediate, since \( V_{\Gamma} \) is built by assigning 1 to every formula in \( \Gamma \) and closing the set of 1-values under \( \uparrow_U \)-rules, following Definitions 8.1–8.2.

\[ \]

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

**Proof.** (\( \neg \)). First, \( V_{\Gamma}(-A) = 1 \) implies \( V_{\Gamma}(A) = 0 \) by Definition 8.2; secondly, an assignment \( V_{\Gamma}(-A) \) occurs only if \( V_{\Gamma}(A) = 1 \), in which case \( V_{\Gamma}(A) = 1 \) by closure under \( E \neg \); finally, the case \( \langle A, \bot \rangle \in V_{\Gamma} \) implies that \( \langle A, 1 \rangle, \langle A, 0 \rangle \not\in V_{\Gamma}^\omega \) and by the above reasoning \( \langle A, 0 \rangle, \langle A, 1 \rangle \not\in V_{\Gamma}^\omega \). Hence \( \langle A, \bot \rangle \in \bot (V_{\Gamma}^\omega) = V_{\Gamma} \). In either case, \( V_{\Gamma}(-A) \in f_\neg (V_{\Gamma}(A)) \).

(\( * \in \{\wedge, \vee, \rightarrow\} \)). 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 \( V_{\Gamma} \) is closed under intelim rules, \( V_{\Gamma}^L = V_{\Gamma}(\mathcal{L} \times \{1, 0, \bot\}) \) is a function over \( \mathcal{L} \) closed under intelim rules over \( \mathcal{L} \). It is also \( \uparrow_0 \)-consistent, since \( V_{\Gamma} \) is \( \uparrow_0 \)-consistent and \( \uparrow_0 \subseteq \uparrow_U \). By Proposition 3.8, \( V_{\Gamma}^L \) is a valuation over \( \mathcal{L} \) and so satisfies the truth-tables over \( \mathcal{L} \). Since \( V_{\Gamma}^L \subseteq V_{\Gamma} \), \( V_{\Gamma} \) also satisfies the truth-tables over \( \mathcal{L} \).

(Ind. case \( B, C \in \mathcal{L}_{U \leq k+1} \).) Suppose as inductive hypothesis that \( V_{\Gamma} \) 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 \( B \iff 1 \) or \( C \iff 1 \).) \( V_{\Gamma} \) closed under \( I \), so \( V_{\Gamma}(B \vee C) = 1 \).

(Case \( B \iff 0 \) or \( C \iff 0 \).) 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, \( V_{\Gamma}(-B) = V_{\Gamma}([-\pm A_1]B') = 1 \). Consider the value of \( B \vee C \):

(Subcase \( B \vee C \iff 1 \).) By \textit{Triv}, \( V_{\Gamma}([\pm A_2]B \vee C) = 1 \), so we can apply:

\[
\neg[\pm A_1]B' \quad [\pm A_2]([\pm A_1]B' \vee [\pm A_2]C') \quad [\pm]E \vee
\]

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

(Subcase \( B \vee C \iff 0 \).) Thus, \( V_{\Gamma}(-B \vee C) = 1 \) and by the intelim rule \( E \vee \) we obtain \( V_{\Gamma}(-C) = 1 \), so \( V_{\Gamma}(C) = 0 \).

(Subcase \( B \vee C \iff \bot \).) 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 \( V_{\Gamma}(-C) = 1 \) and with the case
assumption $V_\Gamma(\neg B) = 1$ and $I \lor$ would give $V_\Gamma(\neg (B \lor C)) = 1$, contradicting the subcase assumption. Thus, it must be that $V_\Gamma(C) = \bot$.

In all subcases, $V_\Gamma(B \lor C) = V_\Gamma(C)$, and so $V_\Gamma$ satisfies the truth-table $f_\lor(0, v) = v$ for the case $B \mapsto 0$.

(Case $B \mapsto \bot$ and $C \mapsto \bot$.) We only need to show that $V_\Gamma(B \lor C) \neq 0$, but this is immediate since otherwise $V_\Gamma(\neg (B \lor C)) = 1$, and then we would have $V_\Gamma(\neg B) = 1$ and finally $V_\Gamma(B) = 0$, contradicting the case assumption.

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

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

Proof. Any $V_\Gamma \neq V_\lambda$ satisfies the truth-tables as shown in Lemma 8.5. To see that $M$ is also closed under updates, let $V_\Gamma^A = V_{\Gamma \cup \{A\}}$. This satisfies (i) $V_\Gamma^A \supseteq V_\Gamma$ and (ii) $V_\Gamma^A(A) = 1$. For any other $V_\Delta$ satisfying the corresponding (i)–(ii), it must be that $V_\Delta[\Gamma \cup \{A\}] = 1$ and so by $\vdash_U$-closure, $V_{\Gamma \cup \{A\}} \supseteq V_\Delta$.

Thus, $V_\Gamma^A$ is minimal with (i)–(ii). Since $V_{\Gamma \cup \{A\}} \in M$, we are done.

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

Proof. Let us prove the contrapositive. Assume $\Gamma \not\vdash_U A$. By Lemma 8.4, $V_\Gamma(A) \neq 1$ and so there is $V \in M$ with $V[\Gamma] = \{1\}$ and $V(A) \neq 1$. By Definition 7.3 and Lemma 8.6, we obtain $\Gamma \not\models_M A$ 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: $V_\Gamma \subseteq V$ iff $V \in S_\Gamma$.

Proof. ($\Rightarrow$). Assume $V_\Gamma \subseteq V$. Since $V_\Gamma[\Gamma] = \{1\}$, it also holds that $V[\Gamma] = \{1\}$. This and $V \in S$ (Fact 7.7) imply $V \in S_\Gamma$. ($\Leftarrow$). Ignoring the trivial case $V = V_\lambda$, 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, $\Gamma \models_S B$; in particular $\Gamma \models_S B$. Hence, the assumption $V \in S_\Gamma$ 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 $V, V' \in S$, $\text{Th}(V) \subseteq \text{Th}(V')$ iff $V \subseteq V'$;
- (from Corollary 4.10) $V_\Gamma = V_{\text{cn}_U(\Gamma)}$, where $\text{cn}_U(\Gamma) = \{A \in \mathcal{L}_U : \Gamma \vdash_U A\}$.

Corollary 8.9. (Correspondence) (i) For any valuation $V' = V_{\text{Th}(V')}$; (ii) for any $\Gamma \subseteq \mathcal{L}_U$, $V_\Gamma = \bigcap S_\Gamma$. 

PROOF. For (i), $V_{\text{Th}(V')}(A) = 1$ iff $\text{Th}(V') \vdash_U A$ (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 $\text{Th}(V')$.

For (ii). $(\sqsubseteq)$ From $V_\Gamma \models V$ iff $V \in S_\Gamma$ (Lemma 8.8), it is immediate that: $V_\Gamma \sqsubseteq \bigcap S_\Gamma$. $(\sqsupseteq)$. Assume $(\bigcap S_\Gamma)(A) = 1$. Since $S$ contains all valuations (Fact 7.7), for any $U$-model $U$, $\Gamma \cap S_\Gamma \subseteq S_\Gamma$ and thus $1 = (\bigcap S_\Gamma)(A) \leq (\bigcup S_\Gamma)(A)$. Hence, $\Gamma \vdash_U A$ for any $U$-model $U$ and so, by definition, $\Gamma \vdash_U A$; then, by completeness $\Gamma \vdash_U A$, which finally gives $V_\Gamma(A) = 1$. Since Lemma 4.3 also holds for $L_U$-valuations, we obtain $V_\Gamma \sqsupseteq \bigcap S_\Gamma$. ■

COROLLARY 8.10. (Canonical is standard) The canonical model is the standard model: $M = S$. The $\sqcap$-minimum valuation is $\bigcap S = V_\emptyset$.

PROOF. The inclusion $M \subseteq S$ follows from $M$ being a $U$-model (Lemma 8.6) and all valuations being elements of the standard model $S$. Corollary 8.9 implies the inclusion $S \subseteq M$ since for every valuation $V'$, $V' = V_{\text{Th}(V')}$ and thus $V' \in M$. $\bigcap S = \bigcap S_\emptyset = V_\emptyset$ 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 $V' \in S$. Consider first the case $V'(A) = 1$. Then,

$$V'^A = V_{\text{Th}(V')}^A = V_{\text{Th}(V') \cup \{A\}} = V_{\text{Cn}_U(\text{Th}(V') \cup \{A\})} = V_{\text{Cn}_U(\text{Th}(V'))} = V_{\text{Th}(V')} = V',$$

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

DEFINITION 8.12. (Translation) We define a translation function $t : L_U \to L$ that deletes all modalities as follows:

$$t(p) = p \quad t(\neg A) = \neg t(A) \quad t(A \ast B) = t(A) \ast t(B) \quad t([\pm A]B) = t(B)$$

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

DEFINITION 8.13. (Two-valued valuations) Define $C \subseteq S$ as the set of valuations $\overline{V} : L_U \to \{1, 0\}$. For $\Gamma \subseteq L_U$, define also $C_\Gamma = \{\overline{V} \in C : \overline{V}[\Gamma] = 1\}$.

LEMMA 8.14. (Classical coincidence) For any formula $A \in L_U$ and valuation $\overline{V} : L_U \to \{1, 0\}$, $\overline{V}(A) = \overline{V}(tA)$. As a consequence, $C_\Gamma = C_{t\Gamma}$. 

P. Pardo
PROOF. By induction on $A$. Given that $V$ is a two-valued valuation, the truth-tables reduce to the classical case, and so the claim holds for boolean operators $A \ast B$ 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. $S$-valuations build from propositional ones in $V$ (Definition 7.6), so one can rephrase classical consequence as: $\Delta \models A$ iff $V(A) = 1$ for all $V \in C_\Delta$. Then, we reason as follows:

$$
\Gamma \models_U B \Rightarrow V(B) = 1 \text{ for all } V \in S_\Gamma \tag{Fact 7.7}
\Rightarrow V(B) = 1 \text{ for all } V \in C_\Gamma \subseteq S_\Gamma
\Rightarrow V(tB) = 1 \text{ for all } V \in C_{t\Gamma} \tag{Lemma 8.14}
\Rightarrow t\Gamma \models tB \quad (t\Gamma \cup \{tB\} \subseteq \mathcal{L}).
$$

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 $\Pi$ be a proof of this. Without loss of generality, assume $\Pi$ contains exactly one instance of $RB$. (Otherwise, isolate a subproof in $\Pi$ with one instance of $RB$, say for $\Delta \vdash A'$ and reason similarly.) From $\Gamma \models_U A$ and $\Gamma \not\models_0 A$, there is some $V \in V_\Gamma$ with $V(A) \neq 1$. As in the proof of Fact 7.7, extend the countermodel $V \in V = S_0$ into a valuation $V'$ in $S$. Obviously, $V'(\Gamma) = 1$ and $V'(A) \neq 1$, so $\Gamma \not\models_S A$, and therefore $\Gamma \not\models_U A$ (contradiction).

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

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

$$
\Gamma \models_{=k} B \text{ iff } \Gamma \models_U [\pm A_1] \ldots [\pm A_k]B \text{ for some } A_1, \ldots, A_k \in f(\Gamma \cup \{B\}).
$$

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 $A_1, \ldots, A_k \in f(\Gamma \cup \{B\})$ and that $V$ ranges over $S_\Gamma$ unless specified otherwise:

$$
\Gamma \models_U [\pm A_1] \ldots [\pm A_{k+1}]B \quad \Leftrightarrow \quad V([\pm A_1] \ldots [\pm A_{k+1}]B) = 1
\Leftrightarrow V([\pm A_1]([\pm A_2] \ldots [\pm A_{k+1}]B)) = 1
\Leftrightarrow V([\pm A_2] \ldots [\pm A_{k+1}]B) = 1 \quad \text{for all } V \in (S_\Gamma)^{A_1} \cup (S_\Gamma)^{\neg A_1}
\Leftrightarrow V([\pm A_2] \ldots [\pm A_{k+1}]B) = 1 \quad \text{for all } V \in S_{\Gamma \cup \{A_1\}} \cup S_{\Gamma \cup \{\neg A_1\}}
\Leftrightarrow \Gamma \cup \{A_1\} \models_U [\pm A_2] \ldots [\pm A_{k+1}]B \quad \Leftrightarrow \Gamma \cup \{A_1\} \models_{=k} B
\Leftrightarrow \Gamma \models_{=k+1} B.
$$
where \((S_\Gamma)^A_1 = S_\Gamma \cup \{A_1\}\) follows from \(V''^{A_1} = V_{\text{th}}^{A_1} = V_{\text{th}}(V') \cup \{A_1\}\) and so \(V'\) is in one set iff it is in the other. (Similarly for \((S_\Gamma)^{\neg A_1} = S_\Gamma \cup \{\neg A_1\}\).) ■

9. Complexity of the Modal Logic \(\mathcal{L}_U\) and Applications

Let us address next the computational complexity of the decision problems \(\text{VAL}_U\) and \(\text{MC}_U^0\) 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 \(\text{VAL}_U \subseteq \mathcal{L}_U\) is defined by: \(A \in \text{VAL}_U\) iff for all \(V \in S\), \(V(A) = 1\). The set \(\text{MC}_U^0 \subseteq \mathcal{P}_\text{fin}(\mathcal{L}_U) \times \mathcal{L}_U \times \{1,0,\perp\}\) is defined by: \((\Gamma, A, v) \in \text{MC}_U^0\) iff \(V_\Gamma(A) = v\). The particular case for \(\Gamma = \emptyset\) is the set denoted \(\text{MC}_U^0\).

**Lemma 9.2.** \(\text{VAL}_U\) is coNP-hard.

**Proof.** We show that the coNP-complete problem of classical validity \(\text{VAL}\) polynomially reduces to \(\text{VAL}_U\). Let \(A \in \mathcal{L}\). For \(\text{at}(A) = \{p_1, \ldots, p_n\}\), 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 \(A \in \text{VAL}\) iff \(f(A) \in \text{VAL}_U\).

**Suppose** \(A \in \text{VAL}\). For \(A \in \text{VAL}_U\), 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 \(B \in \text{sub}(A)\) shows that \(V_\emptyset^\sigma(B) = \mathcal{I}_\sigma(B)\). 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 \(V_\emptyset^\sigma\). Combine this with the assumption \(\mathcal{I}_\sigma(A) = 1\) to conclude that \(V_\emptyset^\sigma(A) = 1\), for any choice \(\sigma\).

Applying Definition 7.2 gives \(V_\emptyset^{p'_1 \cdots p'_{n-1}}([\pm p_n]A) = 1\) for any choice of \(\{p'_1, \ldots, p'_{n-1}\}\) and successive applications render \(V_\emptyset([\pm p_1][\pm p_2] \cdots [\pm p_{n-1}][\pm p_n]A) = 1\). Thus, we proved that \(f(A) \in \text{VAL}_U\).

**Suppose** now that \(A \notin \text{VAL}\). 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, \(f(A) \notin \text{VAL}_U\). ■

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

\[
\begin{align*}
\text{bsub}(p) &= \{p\} \\
\text{bsub}(\neg A) &= \{\neg A\} \cup \text{bsub}(A) \\
\text{bsub}([\pm A]B) &= \{[\pm A]B\} \\
\text{bsub}(A * B) &= \{A * B\} \cup \text{bsub}(A) \cup \text{bsub}(B).
\end{align*}
\]
A Modal View on Resource-Bounded Propositional Logics

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

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

**Theorem 9.4.** $MC^0_U$ is in PSpace. Hence, $\text{Val}_U$ is also in PSpace.

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

- $(\sigma, \neg B)R_0(\sigma, B)$
- $(\sigma, [\pm A_j]B)R_0(\sigma, A_j, B)$
- $(\sigma, B_0 \ast B_1)R_i(\sigma, B_i)$
- $(\sigma, [\pm A_j]B)R_1(\sigma, \neg A_j, B)$

where $i \in \{0, 1\}$, $\ast \in \{\land, \lor, \rightarrow\}$. Note the root is $(\emptyset, A)$ and leaf nodes are of the form $(\sigma, p)$. Our algorithm in Figure 10 traverses (part of) the binary tree $T$ using post-order traversal and evaluates nodes $(\sigma, B) \mapsto (\sigma, B, v(\sigma, B))$ as either $v(\sigma, B) = V_0^\sigma(B)$, or $v(\sigma, B) = \land$ in case $V_0^\sigma$ is 0-depth inconsistent. The algorithm in Figure 10 contains the basic operations:

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

- **0consq(\sigma, B)** in line (2) is also based on [11] and returns $(\sigma, B, v)$ according to: $v = 1$ if $\sigma \vdash_0 B$; $v = 0$ if $\sigma \vdash_0 \neg B$ and $v = \bot$ if $\sigma \not\vdash_0 B$ and $\sigma \not\vdash_0 \neg B$.

- **evaluate(\sigma, \neg B)** in line (2.1) returns $(\sigma, \neg B, f_-(v))$ if the preceding (child) node is $(\sigma, B, v)$.
evaluate($\sigma, B_0 * B_1$) in line (2.1) returns ($\sigma, B_0 * B_1, k_*(v_0, v_1)$) if preceded by ($\sigma, B_0, v_0$) and ($\sigma, B_1, v_1$) \(^{15}\) 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 ($\sigma.A_m, v_0$), ($\sigma.\neg A_m, v_1$):

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

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

(Base case $A = p$.) A quick inspection of $0\text{consq}(\sigma, p)$ (line 2) shows that: $V_\emptyset^\sigma(p) = v$ iff $V_\sigma(p) = v$ iff ($\sigma, p, v$) 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 $V_\emptyset^\sigma(B_0 * B_1)$ does not follow the Kleene truth-table $k_\land(\bot, \bot) = \bot = k_\lor(\bot, \bot)$ is precisely when $\sigma \vdash_0 (\bot)B_0 * B_1$ (line 2). Thus, for all other cases, $f_*$ will coincide with $k_*$, and so $v = V_\emptyset^\sigma(B_0 * B_1)$ will match the node ($\sigma, B_0 * B_1, v$) (line 2.1). Let $\text{trace}(\sigma, A)$ denote the trace of the algorithm for an input node ($\sigma, A$) and let us represent the two stacks in a state by $\langle \alpha \rangle | \langle \beta \rangle$ (or by different columns below):

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

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

P. Pardo
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 $B_0 \ast B_1$. The case for $(σ, ¬B_0)$ is even simpler, as it only generates a left child node.

(Ind. case, $[±C]B$.) Assume the claim for any modal path $σ'$ and a formula $B$. Let $⟨α⟩ | ⟨β⟩$ and $⟨γ⟩ | ⟨δ⟩$ denote states in trace$(σ.C, B)$ resp. trace$(σ.¬C, B)$

\[
\begin{align*}
&\langle(σ, [±C]B)\rangle & \text{—} & \text{initial state} \\
&\langle(σ.C, B), (σ.¬C, B)\rangle & \langle(σ, [±C]B)\rangle & \text{line 3} \\
&\langleα, (σ.¬C, B)\rangle & \langleβ, (σ, [±C]B)\rangle & \text{for all } ⟨α⟩ \mid ⟨β⟩ \\
&\text{—} & \text{—} & \text{(ind. hyp. on } B_0) \\
&\langleγ\rangle & \langleδ, (σ.C, B, v_0), (σ, [±C]B)\rangle & \text{for all } ⟨γ⟩ \mid ⟨δ⟩ \\
&\text{—} & \text{—} & \text{(ind. hyp. on } B_1) \\
&\text{—} & \langle(σ, [±C]B, \text{meet}(v_0, v_1))\rangle & \text{returns this node.}
\end{align*}
\]

In view of Remark 2, an inspection of meet shows that if $V_0^{σ.C}(B) = v_0$ and $V_0^{σ.¬C}(B) = v_1$ (the ind. hyp.), then $V_0^σ([±C]B) = \text{meet}(v_0, v_1)$.

Thus, in all cases the algorithm terminates and returns a sound input $(σ, B, V_0^σ(B))$. In particular, for inputs $(∅, A)$ corresponding to the root of the tree $T$ for $A$, we obtain that $V_0^σ(A) = v$ iff the algorithm returns $(∅, A, v)$.

(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). 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 $|(σ, B)| \leq |A|$. (Time complexity) Observe first that each path consistency$(σ, B)$, each test on $B ∈ L$ and each $0\text{consq}(σ, B)$ operation (lines 1, 2) takes polynomial time $O(|σ \cup \{B\}|^2)$ in $|A|$ (by Proposition 1.10); for any other formula, evaluate$(σ, B)$ 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 $[±A_i]B_i$ doubles the number of required operations upon sub$(B_i)$, namely for the two branches $V_0^{A_i}$ and

\[\text{Moreover, the algorithm might not explore the full tree } T: \text{ at the first sign of } 0\text{-depth inconsistency or } ⊢_0 \text{ consequence at a node } (σ.A_m, B), \text{ the subtree below this node is pruned.}\]
Thus, the number of nodes in the tree is exponential in the length of $A$. This gives an exponential time upper bound.

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

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:

$$BPVAL^f_U = \{[\pm A_1] \ldots [\pm A_k]B \in VAL_U : B \in L \text{ and } A_1, \ldots, A_m \in f(\{B\})\}.$$  

**Theorem 9.6.** $BPVAL^f_U$ is $coNP$-complete.

**Proof.** ($BPVAL^f_U$ is $coNP$-hard.) This can be shown as in the proof that $VAL_U$ is $coNP$-hard (Lemma 9.2): the same reduction $f(A)$ gives atomic blueprints and any virtual space function $f(\cdot)$ contains $at(\cdot)$ (Definition 1.8). ($BPVAL^f_U$ is in $coNP$.) Let $VAL^f_k$ denote the set of formulas valid in the degree-bounded logic $\models^f_k$. Recall that for any virtual space function $f$, $VAL^f_k$ 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 $BPVAL^f_U$ into $VAL$.

$$A \in BPVAL^f_U \text{ iff } A = [\pm A_1] \ldots [\pm A_k]B \text{ for some } k \text{ and } B \in L \text{ and } \{A_1, \ldots, A_k\} \subseteq f(\{B\}) \text{ and } \emptyset \models^f_k [\pm A_1] \ldots [\pm A_k]B$$

$$\text{iff for some } k, \emptyset \models^f_k B \quad \text{(Lemma 8.17)}$$

$$\text{iff for some } k, B \in VAL^f_k \quad \text{(by definition)}$$

$$\text{iff } B \in VAL = \bigcup_k VAL^f_k \quad \text{(Lemma 6.5)}.$$

The same argument shows that deciding whether a proof blueprint is a logical consequence of some propositional database $\Gamma$ 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 $\Gamma$ 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 $SAT_U$ resp. $\bot SAT_U$. Each is defined as the set of formulas $A$ for which there is some $V \neq V_\lambda$ with $V(A) = 1$ and resp. $V(A) = \bot$. While $SAT_U$ can
be shown to be \( NP \)-hard using the same function \( f \) as in Lemma 9.2, \( \perp \text{SAT}_U \) is in \( PSpace \) (by Theorem 9.4 and that \( A \in \perp \text{SAT}_U \) iff \( V_\emptyset(A) = \perp \)).

**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 succinctness. For proof blueprints, generated directly from \( \vdash_U \) or encoding some \( \models_f \) \( k \)-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 \( \Gamma \), and are tasked with answering queries. Say an agent, executing the algorithm for some degree-bounded logic \( \models_f \) \( k \), 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 \( \Gamma \not\models_f B \) does not rule out that \( \Gamma \models_f n B \) 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 \( \{A_1, \ldots, A_m\} \subseteq f(\{B\}) \) 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 \( [\pm A] \ldots [\pm A'][\pm C] \ldots [\pm C']B \land D \) as a generalization of the modal intelim rule \( I[\pm] \land \) (derived using \( \text{Triv} \) and \( \text{Perm} \)); 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 \( \text{RE} \)-equivalent update in the prefix (using \( \text{RE} \));

- **(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 \( \text{RB} \) 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 \( \text{UBBL} \) that approximate classical logic. These logics bound the number of formulas discharged in proofs by the rule of bivalence (\( \text{RB} \)): if \( B \) follows from \( A \) and from \( \neg A \), then \( B \). In a lattice of valuations, we showed that \( \text{RB} \) can be simulated by updates \( V \mapsto V^{[\pm]A} \) resulting in valuations that behave classically w.r.t. \( A \) and thus validate \( A \lor \neg A \). We also proved that updates satisfy permutation and reduce to theory expansions.

A modal logic \( \models_U \) was then presented where formulas \([\pm A]\ldots [\pm A']B\) make explicit \( \text{RB} \)-instances that would suffice in a classical proof of \( B \). The logic thus contains all classical validities prefixed with updates, such as \([\pm p](p \lor \neg p)\), and is a conservative extension of the \( \text{RB} \)-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 \( \text{PSpace} \) upper bound. Towards applications in distributed reasoning, we established the complexity class for validity over proof blueprints as \( \text{coNP} \)-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 \( \text{DBBL} \) 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 \( \text{DBBL} \), one might also ask what modal logic of updates can capture the \( \text{DBBL} \) hierarchy, as \( \models_U \) does for \( \text{UBBL} \). To this end, instead of
representing RB applications in a proof as a sequence $[±A]…[±A']$, the depth-bounded case might use conditional programs $π$ built from: updates $±C$, tests $?C$, composition $π;π'$ and choice $π∪π'$. We conjecture that a PDL-style language [20] with formulas $[π]A$ built from these programs, the formula $[±A;( π)∪(?¬A; π')]B$ 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 $|=U$ with modalities for public announcements of proof blueprints, with formulas $⟨![±A_1]…[±A_k]B⟩C$ expressing: $C$ is known after $[±A_1]…[±A_k]B$ 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.

**Open Access.** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
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 $Y^\ell$;
- the meet of $Y$, denoted $\bigwedge Y$, if $\alpha$ is the greatest lower bound: $\beta \in Y^\ell$ implies $\beta \leq \alpha$. If moreover $\bigwedge Y \in Y$, 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 $\bigvee Y$ or least upper bound of $Y$. In particular, $\bigwedge \emptyset$ is the $\leq$-maximum and $\bigvee \emptyset$ 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: $\bigwedge Y \in X$ whenever $\emptyset \neq Y \subseteq X$. A complete lattice is a poset $(X, \leq)$ closed under arbitrary meets and joins: $Y \subseteq X$ implies $\bigwedge Y, \bigvee Y \in X$.

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 $(X, \leq)$,
$$x \cap y \leq x, y \leq x \sqcup y \quad x \leq y \text{ iff } \uparrow y \subseteq \uparrow x$$
Both $\cap$ and $\sqcup$ are order-preserving: $x \leq y$ and $x' \leq y'$ implies $x \cap x' \leq y \cap y'$; in particular, $x, x' \leq y$ implies $x \cap x' \leq y$. And the same holds for $\sqcup$.

Lemma A.6. (Connecting lemma [16, p.390]) For any lattice $(X, \leq)$,
$$x \cap y = x \quad \text{iff} \quad x \leq y \quad \text{iff} \quad x \sqcup y = y.$$  

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:
$$x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z \quad \text{(assoc.)} \quad x \sqcup y = y \sqcup x \quad \text{(comm.)} \quad x \sqcup x = x \quad \text{(idemp.)} \quad x \cap (y \sqcup x) = x \quad \text{(absorption)}$$
and dual laws hold for meet $\cap$ (i.e. by switching $\cap$ 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:
\[
x \land (y \lor z) \geq (x \land y) \lor (x \land z) \quad \text{and} \quad x \lor (y \land z) \leq (x \lor y) \land (x \lor z).
\]

B. Appendix: Proofs

PROPOSITION 3.4. For any non-empty subset \(\mathcal{V}' \subseteq \mathcal{V}\), the meet exists in \(\mathcal{V}\) as the pointwise meet \(\bigcap \mathcal{V}' = \{\langle A, \bigcap_{V \in \mathcal{V}'} V(A) \rangle : A \in \mathcal{L}\}\).

PROOF. The pointwise meet of a set of \(3\)-valued functions is clearly its greatest lower bound. Let us show then \(\bigcap \mathcal{V}' \in \mathcal{V}\), i.e. that \(\bigcap \mathcal{V}'\) respects the truth-tables. For negation \(\neg\), we omit for all \(V \in \mathcal{V}'\) and reason as follows:
\[
(\bigcap \mathcal{V}')(\neg A) = \begin{cases} 
1 & \text{if } V(\neg A) = 1 \iff V(A) = 0 \iff (\bigcap \mathcal{V}')(A) = 0 \\
0 & \text{if } V(\neg A) = 0 \iff V(A) = 1 \iff (\bigcap \mathcal{V}')(A) = 1 \\
\bot & \text{other cases } \iff \text{other cases } \iff (\bigcap \mathcal{V}')(A) = \bot
\end{cases}
\]
where other cases \(X \subseteq \{1, 0, \bot\}\) in the last \(\iff\) step amount to \(\{0\} \neq X \neq \{1\}\) and so \(\bot = \bigcap X\). For the remaining connectives, we only prove the conjunction case by considering all possible values \(\langle v, v' \rangle = \langle (\bigcap \mathcal{V}')(A), (\bigcap \mathcal{V}')(B) \rangle\):
\[
\langle 0, v' \rangle; \langle v, 0 \rangle. \quad \text{We only check } v = (\bigcap \mathcal{V}')(A) = 0. \quad \text{Then clearly } V(A) = 0 \quad \text{for all } V \in \mathcal{V}' \quad \text{and so } V(A \land B) = 0 \quad \text{for all } V \in \mathcal{V}'. \quad \text{Hence } (\bigcap \mathcal{V}')(A \land B) = 0.
\]
\[
\langle 1, 1 \rangle; \langle \bot, \bot \rangle. \quad \text{Say } (\bigcap \mathcal{V}')(B) = \bot. \quad \text{Then the set of } B\text{-values } X = \{V(B) : V \in \mathcal{V}\} \text{ satisfies: } \{1\} \neq X \neq \{0\} \quad \text{and so } \bigcap X = \bot. \quad \text{Since } f_{\land}(1,v') = v', \text{we also have } X = \{V(A \land B) : V \in \mathcal{V}'\} \text{ which gives } (\bigcap \mathcal{V}')(A \land B) = \bigcap X = \bot.
\]
\[
\langle \bot, \bot \rangle. \quad \text{We check that } (\bigcap \mathcal{V}')(A \land B) \in \{0, \bot\}. \quad \text{Let } X, Y \text{ be the sets of values of } A \text{ resp. } B \text{ among } \mathcal{V}'. \quad \text{Again each set is different from } \{1\} \text{ and } \{0\}, \quad \text{so if } (\bigcap \mathcal{V}')(A \land B) = 1 \text{ the sets would be } X = \{1\} = Y \text{ (contradiction)}. \quad \blacksquare
\]

FACT 3.8. \(\mathcal{V}^+\) is the set of functions \(V : \mathcal{L} \rightarrow \{1, 0, \bot\}\) closed under applications of intelim rules.

PROOF. Given Proposition 3.2, the \((\subseteq)\) direction reduces to checking that \(V_{\land}\) is closed under intelim rules, and indeed all rule conclusions are true in \(V_{\land}\).

For \((\supseteq)\), let \(f'\) be in the set \(\{f : \mathcal{L} \rightarrow \{1, 0, \bot\} \mid f \text{ is closed under intelim rules}\}\).

In case \(f'\) respects the truth-tables, by Definitions 1.2, 3.1 we obtain that \(f' \in \mathcal{V}\) 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 \(f'(A) = 1 \quad \text{and} \quad f'(\neg A) = 1\), or that \((A,v), (A,v') \in f'\) with \(v \neq v'\). In the first case, we must have \(f' = V_{\land}\) since closure under \(I\land\) and \(E\land\) 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, $f' \in V_\lambda$. ■

**Fact 4.2.** For any $V \in \mathbb{V}^+$, $\text{Th}(V)$ is a theory: $\mathbb{C}n_0(\text{Th}(V)) \subseteq \text{Th}(V)$.

**Proof.** From $A \in \mathbb{C}n_0(\text{Th}(V))$, we get $\text{Th}(V) \vdash_0 A$, so by soundness (Proposition 1.11) $\text{Th}(V) \models_0 A$. In particular, if $V[\text{Th}(V)] = \{1\}$ then $V(A) = 1$. But the antecedent is immediate, so $V(A) = 1$ and thus $A \in \text{Th}(V)$. ■

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

**Proof.** ($\Leftarrow$) The assumption gives $V(A) \leq V'(A)$, so if $V(A) = 1$, we also have $V'(A) = 1$. Hence $\text{Th}(V) \subseteq \text{Th}(V')$. ($\Rightarrow$) Assume $\text{Th}(V) \subseteq \text{Th}(V')$ and let $A \in \mathcal{L}$. We show $V(A) \leq V'(A)$. (Case $V(A) = 1$.) Then $V(\neg A) = 1$ and reasoning as in the previous case we obtain $V'(-A) = 1$ 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 $V \subseteq V'$. ■

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

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

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

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

Since $\Gamma$ 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 $V^\omega$. This and the fact that $V_{\Gamma}$ is closed under intelim rules imply that $V_{\Gamma}$ 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, $\text{Th}(V_{\Gamma}) \subseteq \text{Th}(V)$ so by Lemma 4.3, $V_{\Gamma} \subseteq V$ and moreover $V_{\Gamma} \subseteq \prod V_{\Gamma}$. With $V_{\Gamma} \in \mathbb{V}_{\Gamma}$, we conclude $V_{\Gamma} = \prod V_{\Gamma}$. ■

**Fact 4.7.** For any set $\Gamma \subseteq \mathcal{L}$, (i) $\forall \Gamma = \uparrow V_{\Gamma}$ and (ii) $\bigcap \uparrow V_{\Gamma} = V_{\Gamma}$.
PROOF. For (i) let $V' \in \mathcal{V}_\Gamma$ be arbitrary. Then,

\begin{align*}
\text{(i)} & \quad V' \in \mathcal{V}_\Gamma \text{ iff } V'[\Gamma] = \{1\} \quad \text{iff } V_\Gamma \subseteq V' \in \mathcal{V}^+ \\
& \quad \text{iff } V' \in \uparrow V_\Gamma \text{ (by (i)).}
\end{align*}

**Corollary 4.10.** For any $\Gamma \subseteq \mathcal{L}$, (i) $V_\Gamma = V_{\mathcal{C}_n(\Gamma)}$ and (ii) $\mathcal{C}_n(\Gamma) = \text{Th}(V_\Gamma)$.

**Proof.** For claim (i), $\mathcal{L} \subseteq V_{\mathcal{C}_n(\Gamma)}$ follows from $\mathcal{L} \subseteq \mathcal{C}_n(\Gamma)$ by reasoning similarly to the proof of Lemma 4.3($\Rightarrow$). $\mathcal{L}$ Let $A \in \mathcal{L}$ be arbitrary with $V_{\mathcal{C}_n(\Gamma)}(A) = 1$. That is, $\mathcal{C}_n(\Gamma) \models_0 A$, and so by completeness $\mathcal{C}_n(\Gamma) \vdash_0 A$. Since $\mathcal{C}_n(\Gamma)$ is a theory, $A \in \mathcal{C}_n(\Gamma)$ 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 $\text{Th}(V_{\mathcal{C}_n(\Gamma)}) \subseteq \text{Th}(V_\Gamma)$ so Lemma 4.3 gives $V_{\mathcal{C}_n(\Gamma)} \subseteq V_\Gamma$. For claim (ii),

\[ \mathcal{C}_n(\Gamma) = \text{Th}(V_{\mathcal{C}_n(\Gamma)}) \text{ by Prop. 4.9(i)} \]

\[ = \text{Th}(V_\Gamma) \text{ by (i) with } V_{\mathcal{C}_n(\Gamma)} = V_\Gamma. \]

**Fact 4.11.** $\text{Th}(V_1 \cap V_2) = \text{Th}(V_1) \cap \text{Th}(V_2)$. As a consequence, $V_\Gamma \cap V_\Delta = V_{\mathcal{C}_n(\Gamma) \cap \mathcal{C}_n(\Delta)}$.

**Proof.** We reason as follows:

\[
\text{Th}(V_1 \cap V_2) = \{A \in \mathcal{L} : (V_1 \cap V_2)(A) = 1\} = \{A \in \mathcal{L} : V_1(A) = 1\} \cap \{A \in \mathcal{L} : V_2(A) = 1\} = \text{Th}(V_1) \cap \text{Th}(V_2).
\]

\[
\text{Th}(V_\Gamma \cap V_\Delta) = \text{Th}(V_\Gamma) \cap \text{Th}(V_\Delta) \quad (\text{by the above claim}) = \mathcal{C}_n(\Gamma) \cap \mathcal{C}_n(\Delta) \quad (\text{by Fact 4.7(ii)}).
\]

This and Proposition 4.9(i) imply that $V_\Gamma \cap V_\Delta = V_{\text{Th}(V_\Gamma \cap V_\Delta)} = V_{\mathcal{C}_n(\Gamma) \cap \mathcal{C}_n(\Delta)}$.

**Proposition 4.12.** The join $\biguplus \mathcal{V}' = \bigcap (\mathcal{V}')^u$ obtained from Proposition 3.7 is: $\biguplus \mathcal{V}' = V_{\bigcup_{V' \in \mathcal{V}'} \text{Th}(V')}$, and so another application of

\[
(\mathcal{V}')^u = \{V^* : V^* \supseteq V', \text{ for all } V' \in \mathcal{V}'\} = \{V^* : \text{Th}(V^*) \supseteq \text{Th}(V'), \text{ for all } V' \in \mathcal{V}'\} = \{V^* : \text{Th}(V^*) \supseteq \bigcup_{V' \in \mathcal{V}'} \text{Th}(V')\} = \{V^* : V^* \supseteq V_{\bigcup_{V' \in \mathcal{V}'} \text{Th}(V')}\} = \uparrow V_{\bigcup_{V' \in \mathcal{V}'} \text{Th}(V')}.
\]

Hence $\bigcap(\mathcal{V}')^u(A) = 1$ iff $\bigcup_{V' \in \mathcal{V'}} \text{Th}(V')(A) = 1$ iff $V_{\bigcup_{V' \in \mathcal{V}'} \text{Th}(V')}(A) = 1$. That is, $\text{Th}(\bigcap(\mathcal{V}')^u) = \text{Th}(V_{\bigcup_{V' \in \mathcal{V}'} \text{Th}(V')})$ and so another application of
Lemma 4.3 gives \( \prod (\nabla')^a = V_{\cup \nu \in \nu} T_h(\nu') \). For the claim, \( V_\Gamma \uplus V_\Delta = V_{\Gamma \cup \Delta} \).

\[
V_\Gamma \uplus V_\Delta = \bigcup \{V_\Gamma, V_\Delta\} = V_{T_h(\Gamma) \cup T_h(\Delta)} \quad \text{(by the above claim)}
\]

\[
= V_{C_{n_0}(\Gamma) \cup C_{n_0}(\Delta)} \quad \text{(by Coro. 4.10(ii))}
\]

\[
= V_{C_{n_0}(\Gamma \cup \Delta)} \quad \text{(by Coro. 4.10(i))}
\]

\[
= V_{\Gamma \cup \Delta} \quad \text{(since \( \vdash_0 \) is tarskian)}
\]

\[
= V_{\Gamma \cup \Delta} \quad \text{(by Coro. 4.10(i)).} \]

\[
\text{Corollary 5.3.} \quad \text{(i) If } V(A) = 1, \text{ then } V^A = V; \quad \text{(ii) if } V(A) = 0, \text{ then } V^A = V_\lambda.
\]

\[
\text{Proof.} \quad \text{For (i), if either } V \in V \text{ or } V = V_\lambda, \text{ then } V(A) = 1 \text{ implies } V_{\{A\}} \subseteq V. \text{ By Lemma A.6, } V \uplus V_{\{A\}} = V, \text{ so } V^A = V. \text{ For (ii), } V(A) = 0 \text{ and } V_{\{A\}}(A) = 1 \text{ imply } V^A = V \uplus V_{\{A\}} = \{V_\lambda\} \text{ and then } V^A = \prod V^A = \prod \{V_\lambda\} = V_\lambda. \]

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

\[
\text{Proof.} \quad \text{By Corollary 5.3, if } V(A) = 1 \text{ resp. } V(A) = 0, \text{ then } V^{[\pm]A} = V \cap V_\lambda \text{ resp. } V^{[\pm]A} = V_\lambda \cap V. \text{ By Lemma A.6 and } V \subseteq V_\lambda \text{ we get } V \cap V_\lambda = V = V_\lambda \cap V, \text{ so in any case we obtain } V^{[\pm]A} = V. \]

\[
\text{Proposition 5.7.} \quad \text{For any virtual space function } f (\text{Definition 1.8), } \Gamma \uplus f_1 B \text{ iff } V_{\Gamma \{A\}}(B) = 1 \text{ for some } A \in f(\Gamma \cup \{B\}).
\]

\[
\text{Proof.} \quad \text{We use Lemma 5.4 and Corollary 4.8. Let } A \in f(\Gamma \cup \{B\}). \text{ Then,}
\]

\[
V_{\Gamma \{A\}}(B) = 1 \quad \forall (V_\Gamma^A \land V_\Gamma^{-A})(B) = 1 \quad \forall V_\Gamma^A(B) = 1 \quad = V_\Gamma^{-A}(B)
\]

\[
= V_{\Gamma \cup \{\pm A\}}(B) = \Gamma \uplus \{\pm A\} =_0 B \quad \Gamma \uplus f_1 B.
\]

\[
\text{Fact 6.4.} \quad \text{It holds that } \models_1 = \models_1 \text{ and } \models_k \subsetneq \models_k \text{ for any } k > 1.
\]

\[
\text{Proof.} \quad \text{The claim } \models_1 = \models_1 \text{ follows from Proposition 5.7. The inclusion } \models_k \subsetneq \models_k \text{ is a consequence of Proposition 2.2. This inclusion is proper } \models_k \subsetneq \models_k,
\]

\[
\text{as shown by a generalization of the example Table 1(mid) from } k = 2 \text{ to any } k.
\]

\[
\text{Proposition 6.5.} \quad \models = \bigcup_k \models_k^f, \text{ for any } f \text{ as in Definition 1.8.}
\]

\[
\text{Proof.} \quad \text{Assume } \Gamma \models B. \text{ By compactness there is a finite } \Delta \subseteq \Gamma \text{ with } \Delta \models B. \text{ Applying } [10, \text{Prop. 3.2}] \text{ gives us that } \Delta \models_k^f B \text{ for some } k, \text{ so by Proposition 2.2 we obtain } \Delta \models_k^f B \text{ and so } \Gamma \models_k^f B. \]

\[
\text{Proposition 6.6.} \quad \text{For any } \Gamma \uplus \{A, B\} \subseteq \mathcal{L} \text{ with } A \in f(\Gamma \cup \{B\}), \text{ if } \Gamma \models f_i A \text{ and } \Gamma \uplus \{A\} \models f_j B \text{ then } \Gamma \models f_{i+j} B.
\]
Proof. From the assumptions, there are $C_1, \ldots, C_i \in f(\Gamma \cup \{A\})$ and $C_{i+1}, \ldots, C_{i+j} \in f(\Gamma \cup \{A, B\})$ with 

$$\Gamma \cup \{\pm C_1, \ldots, \pm C_i\} \models A$$ and $\Gamma \cup \{A\} \cup \{\pm C_{i+1}, \ldots, \pm C_{i+j}\} \models B. \tag{i}$$

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

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

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

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

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

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

(iii) $V(A) = 1$ for some classical inconsistency $A$, e.g. $\models \neg A$.

Proof. (ii) $\Rightarrow$ (i). Suppose $V[[\pm]A_1, \ldots, [\pm]A_k] = V_\bot$ and, towards a contradiction, that $V \subseteq \overline{V}$ 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, $\overline{V}[[\pm]A_1, \ldots, [\pm]A_k] = \overline{V}$. With this and the assumptions, we obtain $V_\bot = V[[\pm]A_1, \ldots, [\pm]A_k] \subseteq \overline{V}[[\pm]A_1, \ldots, [\pm]A_k] = \overline{V}$ (contradiction).

(iii) $\Rightarrow$ (ii). Assume $V(A) = 1$ with $\models \neg A$ and let $at(\neg A) = \{p, \ldots, p'\}$.

$$\models \neg A \Rightarrow \models at_k \neg A \quad \text{for some } k \leq |\{p, \ldots, p'\}| \tag{Prop. 6.5}$$

$$\Rightarrow V[[\pm]p_1, \ldots, [\pm]p_k](\neg A) = 1 \quad \text{for any } V' \in \mathbb{V} \tag{Def. 7.2}$$

In particular, $V[[\pm]p_1, \ldots, [\pm]p_k](\neg A) = 1$ and since $V \subseteq V[[\pm]p_1, \ldots, [\pm]p_k]$, the assumption $V(A) = 1$ implies $V[[\pm]p_1, \ldots, [\pm]p_k](A) = 1$. Using Fact 3.8, we obtain $V[[\pm]p_1, \ldots, [\pm]p_k](B) = 1$ for any $B \in \mathcal{L}$. That is, $V[[\pm]p_1, \ldots, [\pm]p_k] = V_\bot$.

(i) $\Rightarrow$ (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

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

The inductive step $V^n \rightarrow V^{n+1}$ clearly preserves classical consistency and membership in $\mathbb{V}$. As a consequence, also $V^\omega = \bigcup_n V^n$ is classically consistent ($V^\omega(A) \neq 1$ whenever $\models \neg A$) and a valuation in $\mathbb{V}$. The latter, together with $V(p_i) \in \{1, 0\}$ for any $p_i \in \mathbb{Var}$, implies that $V \subseteq V^\omega \in \mathbb{C}. \tag{iii}$
P. Pardo

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

[5] Bjerring, J., 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. Hory, 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.
P. Pardo

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

[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] Schaefer, 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.