Tractable depth-bounded approximations to FDE and its satellites

MARCELLO D'AGOSTINO, *Department of Philosophy, University of Milan, Milano 20122, Italy.*
E-mail: marcello.dagostino@unimi.it

ALEJANDRO SOLARES-ROJAS, *Department of Philosophy, University of Milan, Milano 20122, Italy.*
E-mail: alejandro.solares@unimi.it

To the memory of Carlos César Jiménez

Abstract

FDE, **LP** and $\mathbf{K_3}$ are closely related to each other and admit of an intuitive informational interpretation. However, all these logics are co-NP complete, and so idealized models of how an agent *can* think. We address this issue by shifting to signed formulae, where the signs express imprecise values associated with two bipartitions of the corresponding set of standard values. We present proof systems whose operational rules are all linear and have only two structural branching rules that express a generalized Principle of Bivalence. Each of these systems leads to defining an infinite hierarchy of tractable approximations to the respective logic, in terms of the maximum number of allowed nested applications of the two branching rules. Further, each resulting hierarchy admits of an intuitive 5-valued non-deterministic semantics.

Keywords: **FDE**, **LP**, $\mathbf{K_3}$, tractability, natural deduction, tableaux, non-deterministic semantics

1 Introduction

Many interesting propositional logics are likely to be intractable. For instance, Classical Propositional Logic (**CPL**), First-Degree Entailment (**FDE**) [1], the Logic of Paradox (**LP**) [6, 63] and Strong Kleene Logic ($\mathbf{K_3}$) [50] are all co-NP complete [see 3, 24, 75]. Thus, we cannot expect a real agent, no matter whether human or artificial, to be always able to recognize in practice that a certain conclusion follows from a given set of assumptions. This is a source of major difficulties in research areas that are in need of less idealized, yet theoretically principled, models of logical agents with bounded cognitive and computational resources. The 'depth-bounded approach' to **CPL** [e.g. 30–32] provides an account of how this logic can be approximated in practice by realistic agents in two moves: (i) by providing a semantic and proof-theoretic characterization of a tractable 0-depth approximation and (ii) by defining an infinite hierarchy of tractable k-depth approximations, which can be naturally related to a hierarchy of realistic, resource-bounded agents, and admits of an elegant proof-theoretic characterization.

A key idea underlying the 'depth-bounded approach' to **CPL** is that the meaning of a logical operator is specified solely in terms of the information that is actually possessed by an agent, i.e. information practically accessible to her and with which she can operate. This kind of information is called *actual*, and we use the verb 'to hold' as synonymous with 'to actually possess'. The semantics is ultimately based on intuitive, albeit non-deterministic, 3-valued tables that were first

Vol. 34, No. 5,
Advance Access Publication on 19 June 2023 https://doi.org/10.1093/logcom/exad040

put forward by Quine [66] to capture the 'primitive meaning of the logical constants' [see also 25], where the values have a natural informational interpretation ('assent', 'dissent', 'abstain'). The proof-theoretic characterization given in [30, 31] is based on introduction and elimination (intelim) rules that, unlike those of Gentzen-style natural deduction, involve no 'discharge' of hypotheses. The *0-depth approximation* consists of the consequence relation associated with the intelim rules only, is computationally easy (tractable) and corresponds to Quine's non-deterministic semantics. The *depth* of **CPL** inferences is measured in terms of the maximum number of nested applications of a single branching rule, which is a Classical Dilemma rule called *PB* ('Principle of Bivalence'). *PB* governs the manipulation of *virtual* information, i.e. hypothetical information that an agent does not hold, but he temporarily assumes as if he held it. Intuitively, the more times such virtual information needs to be invoked via *PB*, the harder the corresponding inference is for any agent who is able to perform at least 0-depth inferences, both from the computational and the cognitive viewpoint. Thus, the nested applications of that rule provide a sensible measure of inferential depth. In essence, each k-depth logic corresponds to a limited capability of manipulating virtual information.[1] By contrast, in Gentzen-style proof systems, some of the 'discharge' rules of natural deduction, as well as their counterparts in the sequent calculus, make essential use of virtual information. Given that in Gentzen-style systems cut is eliminable, no hierarchy of approximations can be defined by limiting the application of the cut rule.

The depth-bounded approach to **CPL**, as remarked in [30], is the first step of a more general research program that aims to define similar approximations to first-order logic and to a variety of non-classical logics. A preliminary step of the first order case can be found in [35]. In this paper, we show how the depth-bounded approach can be naturally extended to useful many-valued logics such as **FDE**, **LP** and $\mathbf{K_3}$, which are closely related to each other. The trio of many-valued logics addressed in this paper provides a case study for extending the depth-bounded framework to a variety of finite-valued logics, in the spirit of [20, 21, 49, 62].

The rest of the paper is organized as follows. Section 2 provides some working definitions. Section 3 succinctly recalls **FDE** as interpreted in informational terms, and points out the need of *imprecise values* under such an interpretation. In section 4, we introduce a proof system for **FDE** that will serve as a basis for defining its depth-bounded approximations. Section 5 is devoted to show the subformula property of such a system. In section 6, we define a hierarchy of depth-bounded approximations to **FDE**, and show that each level of it is tractable. Then, in section 7, we provide a 5-valued non-deterministic semantics for that hierarchy, and provide the corresponding soundness and completeness proofs. Next, in section 8, we show that by making minor modifications we obtain analogous hierarchies of tractable depth-bounded approximations to **LP** and $\mathbf{K_3}$. Finally, section 9 recalls a cluster of ideas and proof systems closely related to our approach in this paper.

2 Preliminaries

Let $\mathcal{L}$ denote some *propositional language*; namely, a structure consisting of a countable set of propositional variables, and a finite set $\mathcal{C}(\mathcal{L})$ of connectives, each having a specific natural number as its arity. 0-ary connectives are called *propositional constants*. In turn, let $F(\mathcal{L})$ and $At(\mathcal{L})$ respectively be the set of well-formed and atomic formulae of $\mathcal{L}$. We use $p, q, r, \ldots$, possibly with subscripts, as metalinguistic variables for atomic $\mathcal{L}$-formulae; $A, B, C, \ldots$, possibly with subscripts, for arbitrary

[1] From this point of view, there is a close connection between the intelim system presented in [30, 31], and the systems *KE* [37], *KI* [57] as well as the so-called Stålmarck's method [69]. See section 4 of [31] for more details.

$\mathcal{L}$-formulae; and $\Gamma, \Delta, \Lambda, \ldots$, possibly with subscripts, to vary over sets of $\mathcal{L}$-formulae.[2] Finally, we shall assume that all propositional languages share the same set of atomic variables and, so, we shall identify a language $\mathcal{L}$ with $\mathcal{C}(\mathcal{L})$.

DEFINITIONS 1
For every formula A, a *subformula* of A is defined inductively: (i) A is a subformula of A; (ii) for every binary operator $\circ$, if $B \circ C$ is a subformula of A, then so are B and C; (iii) if $\neg B$ is a subformula of A, so is B; (iv) nothing else is a subformula of A. In turn, a *proper subformula* of A is any subformula of A that is different from A. An *immediate subformula* of A is any proper subformula of A that is not a proper subformula of any proper subformula of A.

NOTATION 2
We denote by sub the function that maps any given set Γ of formulae to the set of all its subformulae, and by at the function that maps any given Γ to the set of its atomic subformulae.

Moreover, we define the *degree* of a $\mathcal{L}$-formula A as the number of occurrences of connectives in A.

Now, in the context of this paper we shall abuse standard terminology and call *consequence relation* on a language $\mathcal{L}$ any relation $\mid\!\sim\, \subseteq 2^{F(\mathcal{L})} \times F(\mathcal{L})$, satisfying the following conditions:

Reflexivity: If $A \in \Gamma$, then $\Gamma \mid\!\sim A$.

Monotonicity: If $\Gamma \mid\!\sim A$, then $\Gamma \cup \Delta \mid\!\sim A$.

In turn, a *Tarskian consequence relation* (Tcr for short) on $\mathcal{L}$ is a consequence relation on $\mathcal{L}$ satisfying the following additional condition:

Cut for sets: If $\Gamma \mid\!\sim A$ for every $A \in \Delta$ and $\Gamma \cup \Delta \mid\!\sim B$, then $\Gamma \mid\!\sim B$.

An $\mathcal{L}$-*substitution* is a function $\sigma : F(\mathcal{L}) \longrightarrow F(\mathcal{L})$ such that for every n-ary connective $\diamond$ and formulae $A_1, \ldots, A_n$,

$$\sigma(\diamond(A_1, \ldots, A_n)) = \begin{cases} \diamond(\sigma(A_1), \ldots, \sigma(A_n)) & \text{if } n > 0 \\ \diamond & \text{if } n = 0 \end{cases}$$

A *Tarskian propositional logic* is a pair $\mathbf{L}=\langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$, where $\mathcal{L}$ is a propositional language and $\mid\!\sim_{\mathbf{L}}$ is a Tcr on $\mathcal{L}$ satisfying the following additional condition:

Structurality: If $\Gamma \mid\!\sim_{\mathbf{L}} A$, then $\sigma(\Gamma) \mid\!\sim_{\mathbf{L}} \sigma(A)$ for every $\mathcal{L}$-substitution σ.[3]

DEFINITIONS 3
Let $\mid\!\sim$ be a Tcr for $\mathcal{L}$. $\mid\!\sim$ is *finitary* if for every Γ and every A such that $\Gamma \mid\!\sim A$, there is a finite $\Delta \subseteq \Gamma$ such that $\Delta \mid\!\sim A$. In turn, a Tarskian propositional logic $\langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$ is *finitary* if so is $\mid\!\sim_{\mathbf{L}}$.

In a finitary Tarskian propositional logic, the following 'restricted' version of transitivity suffices:

Cut (*Transitivity*): If $\Gamma \mid\!\sim A$ and $\Gamma \cup \{A\} \mid\!\sim B$, then $\Gamma \mid\!\sim B$.

[2] For readability, we shall omit the prefix $\mathcal{L}$ in '$\mathcal{L}$-formula(e)', leaving the propositional language at issue implicit in the context.

[3] Where $\sigma(\Gamma)$ is short for $\{\sigma(A) \mid A \in \Gamma\}$.

Since finitariness is essential for *practical* reasoning—where a conclusion is always derived from a finite set of assumptions—here we are interested only in finitary logics.

In turn, the following generalization of the notion of *many-valued matrix*, as well as notions thereof, are essential for our investigation and has been extensively studied by Avron and co-authors [e.g. 2, 8–11]:

DEFINITION 2.1

A *non-deterministic matrix* (*Nmatrix*) for $\mathcal{L}$ is a triple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:

- $\mathcal{V}$ is a non-empty set of truth-values;
- $\mathcal{D}$ is a non-empty proper subset of $\mathcal{V}$ (whose elements are called the *designated* elements of $\mathcal{V}$);
- $\mathcal{O}$ is a function that associates an n-ary function $\widetilde{\diamond} : \mathcal{V}^n \longrightarrow 2^{\mathcal{V}} \backslash \{\emptyset\}$ with every n-ary connective $\diamond$ of $\mathcal{L}$.

We say that $\mathcal{M}$ is (*in*)*finite* if so is $\mathcal{V}$.

DEFINITIONS 4

Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$.

- A *partial $\mathcal{M}$-valuation* for $\mathcal{L}$ is a function $v : F(\mathcal{L})^* \longrightarrow \mathcal{V}$ for some $F(\mathcal{L})^* \subseteq F(\mathcal{L})$ satisfying the following conditions:
  - The set $F(\mathcal{L})^*$ is closed under subformulae; i.e. $\mathsf{sub}(F(\mathcal{L})^*) = F(\mathcal{L})^*$.
  - For each n-ary connective $\diamond$ of $\mathcal{L}$, the following holds for all $A_1, \ldots, A_n \in F(\mathcal{L})^*$:

$$v(\diamond(A_1, \ldots, A_n)) \in \widetilde{\diamond}(v(A_1), \ldots, v(A_n)). \quad (1)$$

- A partial $\mathcal{M}$-valuation is a (full) $\mathcal{M}$-*valuation* if its domain is $F(\mathcal{L})$.

REMARK 2.2

When taking an Nmatrix as a generalization of a matrix, the latter is viewed as a special type of an Nmatrix in which each $\widetilde{\diamond}$ always returns a singleton. In such a case, each $\widetilde{\diamond}$ can be treated as a function $\widetilde{\diamond} : \mathcal{V}^n \longrightarrow \mathcal{V}$. Thus, when there is no risk of confusion, we shall identify singletons of truth-values with the truth-values themselves.

DEFINITIONS 5

Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$. The *Tcr induced by* $\mathcal{M}$, $\vDash_{\mathcal{M}}$, is defined by: $\Gamma \vDash_{\mathcal{M}} A$ if for every partial $\mathcal{M}$-valuation v, if $v(B) \in \mathcal{D}$ for all $B \in \Gamma$, then $v(A) \in \mathcal{D}$. We denote by $\mathbf{L}_{\mathcal{M}} = \langle \mathcal{L}, \vDash_{\mathcal{M}} \rangle$ the *Tarskian propositional logic induced by* $\mathcal{M}$.

DEFINITIONS 6

Given a Tarskian propositional logic $\mathbf{L} = \langle \mathcal{L}, \mid\!\sim_{\mathbf{L}} \rangle$ and a Nmatrix $\mathcal{M}$ for $\mathcal{L}$, we say that:

- $\mathbf{L}$ is *sound* for $\mathcal{M}$ iff $\Gamma \mid\!\sim_{\mathbf{L}} A$ implies $\Gamma \vDash_{\mathcal{M}} A$;
- $\mathbf{L}$ is *complete* for $\mathcal{M}$ iff $\Gamma \vDash_{\mathcal{M}} A$ implies $\Gamma \mid\!\sim_{\mathbf{L}} A$;
- $\mathbf{L}$ is *characterized* by $\mathcal{M}$ iff $\mathbf{L}$ is both sound and complete for $\mathcal{M}$.

The semantics of Nmatrices shares with the semantics of ordinary (deterministic) matrices important properties such as *compactness* [10], *decidability* [see 2] and, even more importantly, *analyticity* [9]. In fact, because of analyticity $\Gamma \vDash_{\mathcal{M}} A$ is decidable whenever Γ and $\mathcal{M}$ are finite. Namely, when assesing whether or not $\Gamma \vDash_{\mathcal{M}} A$, analyticity allows the search to be restricted to partial $\mathcal{M}$-valuations whose domain is $\mathsf{sub}(\Gamma \cup \{A\})$. This, together with the finiteness of Γ and $\mathcal{M}$, assures that the search space is also finite and, thus, that the corresponding algorithm always terminates.[4]

Now, the following notion is fundamental throughout our investigation:

DEFINITION 2.3
Let **L** be a finitary Tarskian propositional logic. An *approximation system* for **L** is a triple $\mathscr{A} = \langle P, \preceq, \{R_\alpha\}_{\alpha \in P}\rangle$, where $\langle P, \preceq\rangle$ is a directed set, called the *parameter set*, and $\{R_\alpha\}_{\alpha \in P}$ is a family of consequence relations on $\mathcal{L}$ such that:

- $\langle P, \preceq\rangle$ has a minimum element 0;
- R_0 is a finitary Tcr;
- $\alpha \prec \beta$ implies $R_\alpha \subset R_\beta$;
- for each $\alpha \in P$, R_α is decidable in polynomial time;
- $\bigcup_{\alpha \in P} R_\alpha = \vdash_{\mathbf{L}}$.

We shall call **L** the *limiting logic* of the approximation system $\mathscr{A}$, and $\langle \mathcal{L}, R_0\rangle$ its *base logic*. Each relation R_α is an *approximation* to **L**.

Naturally, approximation systems are of practical and theoretical interest whenever the limiting logic is known or conjectured to be intractable.[5]

3 Belnap's semantics and the need for imprecise values

First Degree Entailment (**FDE**) captures relevant entailment between implication-free formulae. Based on work of Dunn [e.g. 42], and an observation by Smiley (in correspondence), Belnap [14, 15] gave an interesting semantic characterization of **FDE** in terms of a 4-valued logic, and pointed out its usefulness as the logic in which 'a computer *should* think'. This characterization has become not only the standard semantics of **FDE**, but also its standard presentation. It is motivated from the use of deductive reasoning as a basic tool in the area of 'intelligent' database management or question-answering systems. Databases have a great propensity to be incomplete and become inconsistent: what is stored in a database is usually obtained from different sources which may provide only partial information and may well conflict with each other. For a matrix to characterize a logic adequate for making deductions with information that might be both inconsistent and partial, at least 4 different values are needed [see 2]. An elegant 4-valued matrix is precisely Belnap-Dunn's.

The set of truth-values is $\{\mathbf{t}, \mathbf{f}, \mathbf{b}, \mathbf{n}\}$ and is denoted by $\mathbf{4}$. These values are interpreted as four possible ways in which an atom p can belong to the present state of information of a computer's database, which in turn is fed by a set Ω of equally 'reliable' sources: $\mathbf{t}$ means that the computer is told that p is true by some source, without being told that p is false by any source; $\mathbf{f}$ means that the computer is told that p is false but never told that p is true; $\mathbf{b}$ means that the computer is told

[4] To simplify reading, in what follows we shall omit the prefix or subscript '$\mathcal{M}$' in the notions above.

[5] There is nothing apparent that prevents us to define hierarchies of tractable depth-bounded approximations to finitary non-Tarskian logics. However, we leave that task for future research and so restrict the definition at issue to Tarskian ones.

TABLE 1. **FDE**-tables

$\widetilde{\vee}$	**t**	**f**	**b**	**n**
t	**t**	**t**	**t**	**t**
f	**t**	**f**	**b**	**n**
b	**t**	**b**	**b**	**t**
n	**t**	**n**	**t**	**n**

$\widetilde{\wedge}$	**t**	**f**	**b**	**n**
t	**t**	**f**	**b**	**n**
f	**f**	**f**	**f**	**f**
b	**b**	**f**	**b**	**f**
n	**n**	**f**	**f**	**n**

$\widetilde{\neg}$	
t	**f**
f	**t**
b	**b**
n	**n**

that p is true by some source and that p is false by some other source (or by the same source in different times); **n** means that the computer is told nothing about the value of p. In essence, each value represents a subset of the set {*true*, *false*} of the classical values [42]. These four values form two distinct lattices, depending on whether we consider the partial information ordering induced by set-inclusion (*approximation* lattice) or the partial ordering based on 'closeness to the truth' (*logical* lattice). The information ordering is the one according to which the epistemic state of the computer concerning an atom can evolve over time. As Belnap points out:

> When an atomic formula is entered into the computer as either affirmed or denied, the computer modifies its current set-up by adding a 'told True' or 'told False' according as the formula was affirmed or denied; it does not subtract any information it already has [...] In other words, if p is affirmed, it marks p with **t** if p were previously marked with **n**, with **b** if p were previously marked with **f**; and of course leaves things alone if p was already marked either **t** or **b** [14, p. 12].

A *set-up* is simply an assignment to each of the atoms of exactly one of the values in **4**. The values of complex formulae are obtained by means of considerations related to 'Scott's thesis' about approximation lattices [14], resulting in the truth-tables in Table 1. Using these truth-tables, every set-up can be extended to a valuation function $v : F(\mathcal{L}) \longrightarrow \mathbf{4}$, where $\mathcal{L} = \{\vee, \wedge, \neg\}$, in the usual inductive way. We call this function a **4**-*valuation*. It establishes how the computer is to answer questions about complex formulae based on a set-up. While answering questions on the basis of a given epistemic set up is computationally easy, we do not have a logic yet. As Belnap puts it, we 'want some rules for the computer to use in generating what it implicitly knows from what it explicitly knows', i.e. we need a logic for the computer to reason. This is achieved by turning the truth-tables into a valuation system: Belnap-Dunn's matrix, $\mathcal{M}_\mathbf{4}$, is the matrix for $\mathcal{L}$ where $\mathcal{V} = \mathbf{4}$, $\mathcal{D} = \{\mathbf{t}, \mathbf{b}\}$, and the functions in $\mathcal{O}$ are defined by the truth-tables in Table 1. (Warning: do not confuse the values **t** and **f** in **4** with *true* and *false*. The latter are *local* values referring to the information coming from a source, the former are *global* values, summarinzing the epistemic state of the computer with respect to all the sources.) Thus:

DEFINITION 3.1
A **4**-*valuation* is a function $v : F(\mathcal{L}) \longrightarrow \mathbf{4}$ such that for all A, B:

1. $v(\neg A) = \widetilde{\neg}(v(A))$;
2. $v(A \circ B) = \widetilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$ or $\wedge$.

The consequence relation is then defined as follows:

DEFINITION 3.2
$\Gamma \vDash_{\mathcal{M}_4} A$ iff for every **4**-valuation v, if $v(B) \in \{\mathbf{t}, \mathbf{b}\}$ for all $B \in \Gamma$, then $v(A) \in \{\mathbf{t}, \mathbf{b}\}$.

For the unrestricted language allowing arbitrary formulae involving $\wedge, \vee$ and $\neg$, the decision problem for this consequence relation is co-NP complete [see 3, 75]. This fact follows from the celebrated result by Cook [24] showing that **CPL** is co-NP complete, together with the fact that the decision problem of inconsistency in **CPL** can be reduced to the decision problem of entailment in **FDE**. The latter specifically as follows:

PROPOSITION 3.3
Γ is classically inconsistent iff $\Gamma \vDash_{\mathcal{M}_4} (p_1 \wedge \neg p_1) \vee (p_2 \wedge \neg p_2) \vee ... \vee (p_n \wedge \neg p_n)$, where $p_1, ..., p_n$ are the atoms occurring in Γ.

PROOF. By definition, Γ is classically inconsistent iff there is no classical valuation, $v : F(\mathcal{L}) \longrightarrow \{true, false\}$, such that $v(A) = true$ for all $A \in \Gamma$. In turn, also by definition, this holds iff for every **4**-valuation, $v : F(\mathcal{L}) \longrightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{b}, \mathbf{n}\}$, such that $v(A) \in \{\mathbf{t}, \mathbf{b}\}$ for all $A \in \Gamma$, $v(A) = \mathbf{b}$ for some $A \in \Gamma$ and so, by the **FDE**-tables, $v(p_i) = \mathbf{b}$ for some p_i occurring in Γ. Thus, by the **FDE**-tables for $\neg$ and $\wedge$, this holds iff for every **4**-valuation v such that $v(A) \in \{\mathbf{t}, \mathbf{b}\}$ for all $A \in \Gamma$, $v(p_i \wedge \neg p_i) = \mathbf{b}$ for some p_i occurring in Γ and so, by the **FDE**-table for $\vee$, $v((p_1 \wedge \neg p_1) \vee ... \vee (p_n \wedge \neg p_n)) \in \{\mathbf{t}, \mathbf{b}\}$. Hence, the latter holds iff $\Gamma \vDash_{\mathcal{M}_4} (p_1 \wedge \neg p_1) \vee ... \vee (p_n \wedge \neg p_n)$. □

This situation brings us to the need for tractable approximations. In the next section, we shall present a sort of natural deduction system for **FDE** based on two key observations.

First, as is implicit in the quotation from Belnap above, the values in **4**, except for **b**, cannot be taken as *stable*. An epistemic set up is just a snapshot of an epistemic state that evolves over time. If we want to consider the truth-values **t**, **f**, **n** as stable we need to assume complete information about the set of sources Ω. Namely, while the meaning of **b** is '*there is* at least a source assenting to p and at least a source dissenting from p' (which is information empirically accessible to x in the sense that x may *hold* this information without a complete knowledge of Ω), the meaning of **t**, **f** and **n** involves information of the kind '*there is no* source such that ...', and so requires complete information about the sources in Ω, which may not be empirically accessible to x at any given time. What if the agent does not have such a complete knowledge about the sources? For instance, the agent may well be receiving information from an'open' set of sources as they become accessible (even if the information coming from each single source is assumed to be robust). In such a case, the possibility for an agent to come across a source falsifying 'there is no source such that ...' is always open. Thus, despite their informational nature, three of the values in **4** are *information-transcendent* when interpreted as timeless. They refer to an 'objective' informational situation concerning the domain of all sources, that may well be inaccessible to the computer at any given time. This motivates the need for *a stable imprecise value* such as '**t** *or* **b**', which is implicit in the choice of the set of designated values by Belnap. Inspired by work of D'Agostino [26], and Fitting and Avron [7, 44, 45], we shall address this question by shifting to *signed formulae*, where the signs express such imprecise values associated with two distinct bipartitions of **4**.

A *second* key observation is that, as suggested by Belnap [14, 15], there is no reason to assume that an agent is 'told' about the values of *atoms only*. As we shift from objective truth and falsity to informational truth and falsity, this is a highly unrealistic restriction. In most practical contexts, we may be told that a certain disjunction is true without being told which of the two disjuncts is the true one, or that a certain conjunction is false without being told which of the two conjuncts is the false one. As a simple example of the former situation, take the information that Alice and

Bob are siblings (either they have the same mother or they have the same father); for the latter, take the information that Alice and Bob are not siblings, i.e. for any individual x, the conjunction 'x is a parent of Bob and x is a parent of Alice' must be false, which amounts to saying that either the first or the second conjunct is false, without necessarily knowing which. In the context of **CPL**, these considerations naturally lead to a non-deterministic 3-valued semantics that was anticipated by Quine. See [29] for further references and a discussion that includes the following quotation from Dummett to the effect that in non-mathematical contexts our information may well be *irremediably disjunctive* in nature:

> I may be entitled to assert '*A* or *B*' because I was reliably so informed by someone in a position to know, but if he did not choose to tell me which alternative held good, I could not apply an or-introduction rule to arrive at that conclusion. [...] Hardy may simply not have been able to hear whether Nelson said 'Kismet hardy'or 'Kiss me Hardy', though he heard him say one or the other: once we have the concept of disjunction, our perceptions themselves may assume an irremediably disjunctive form. [...]
> Unlike mathematical information, empirical information decays at two stages: in the process of acquisition, and in the course of retention and transmission. An attendant directing theatre-goers to different entrances according to the colours of their tickets might even register that a ticket was yellow or green, without registering which it was, if holders of tickets of either colours were to use the same entrance; even our observations are incomplete, in the sense that we do not and cannot take in every detail of what is in our sensory fields. That information decays yet further in memory and in the process of being communicated is evident. In mathematics, any effective procedure remains eternally available to be executed; in the world of our experience, the opportunity for inspection and verification is fleeting [41, pp. 266-278].

These two observations prompt us to propose a proof-theoretic approach to depth-bounded reasoning in **FDE** that is similar to the one taken in [30–32] for **CPL**. Before addressing this issue, however, we shall provide in the next section a proof-theoretic characterization of unbounded reasoning in **FDE** that will pave the way for defining its tractable approximations.

4 Intelim deduction in FDE

In what follows we shall use *signed formulae* (*S-formulae* for short). These are expressions of the form $\mathsf{T}A$, $\mathsf{F}A$, T^*A, F^*A, where A is an unsigned formula. Denoting an agent with x and a **4**-valuation with v, their intended interpretation is respectively as follows: 'x holds that A is at least true' (expressing that $v(A) \in \{\mathbf{t}, \mathbf{b}\}$); '$x$ holds that A is non-true' ($v(A) \in \{\mathbf{f}, \mathbf{n}\}$); '$x$ holds that A is non-false' ($v(A) \in \{\mathbf{t}, \mathbf{n}\}$); '$x$ holds that A is at least false' ($v(A) \in \{\mathbf{f}, \mathbf{b}\}$). Crucially, S-formulae of the form $\mathsf{T}A$ or F^*A express information that x may hold even without a complete knowledge of the set sources Ω. However, this is not the case of the other two types of S-formulae which involve complete knowledge of Ω and so can only be assumed hypothetically. Now, we say that the *conjugate* of $\mathsf{T}A$ is $\mathsf{F}A$ and vice versa, and that the conjugate of T^*A is F^*A and vice versa. Let us use S as a variable ranging over $\{\mathsf{T}, \mathsf{F}, \mathsf{T}^*, \mathsf{F}^*\}$, and with $\bar{\mathsf{S}}$ denote: F if $\mathsf{S} = \mathsf{T}$, T if $\mathsf{S} = \mathsf{F}$, F^* if $\mathsf{S} = \mathsf{T}^*$, and T^* if $\mathsf{S} = \mathsf{F}^*$. Besides, we shall write $\mathsf{T}\Gamma$ for $\{\mathsf{T}A \mid A \in \Gamma\}$. Moreover, we shall use $\varphi, \psi, \theta, \ldots$, possibly with subscripts, as variables ranging over S-formulae; and $X, Y, Z, \ldots$, possibly with subscripts, as variables ranging over sets of S-formulae. Further, let us use $\bar{\varphi}$ to denote the conjugate of φ. Finally, we say that the *unsigned part* of an S-formula is the unsigned formula that

TABLE 2. Introduction rules for the standard **FDE** connectives

$$\frac{\mathsf{F}\,A}{\mathsf{F}\,A \wedge B} \qquad \frac{\mathsf{F}\,B}{\mathsf{F}\,A \wedge B} \qquad \frac{\mathsf{F}^*\,A}{\mathsf{F}^*\,A \wedge B} \qquad \frac{\mathsf{F}^*\,B}{\mathsf{F}^*\,A \wedge B}$$

$$\frac{\mathsf{T}\,A}{\mathsf{T}\,A \vee B} \qquad \frac{\mathsf{T}\,B}{\mathsf{T}\,A \vee B} \qquad \frac{\mathsf{T}^*\,A}{\mathsf{T}^*\,A \vee B} \qquad \frac{\mathsf{T}^*\,B}{\mathsf{T}^*\,A \vee B}$$

$$\frac{\begin{array}{c}\mathsf{T}\,A\\ \mathsf{T}\,B\end{array}}{\mathsf{T}\,A \wedge B} \qquad \frac{\begin{array}{c}\mathsf{F}\,A\\ \mathsf{F}\,B\end{array}}{\mathsf{F}\,A \vee B} \qquad \frac{\begin{array}{c}\mathsf{T}^*\,A\\ \mathsf{T}^*\,B\end{array}}{\mathsf{T}^*\,A \wedge B} \qquad \frac{\begin{array}{c}\mathsf{F}^*\,A\\ \mathsf{F}^*\,B\end{array}}{\mathsf{F}^*\,A \vee B}$$

$$\frac{\mathsf{T}\,A}{\mathsf{F}^*\,\neg A} \qquad \frac{\mathsf{F}\,A}{\mathsf{T}^*\,\neg A} \qquad \frac{\mathsf{T}^*\,A}{\mathsf{F}\,\neg A} \qquad \frac{\mathsf{F}^*\,A}{\mathsf{T}\,\neg A}$$

results from it by removing its sign. Given an S-formula φ, we denote by φ^u the unsigned part of φ and by X^u the set $\{\varphi^u \mid \varphi \in X\}$. Note also that, for the reasons explained in the previous section, an agent may hold the information that $\mathsf{T}\,A \vee B$, but neither the information that $\mathsf{T}\,A$ nor that $\mathsf{T}\,B$. Similarly, she may hold the information that $\mathsf{F}^*\,A \wedge B$, but neither the information that $\mathsf{F}^*\,A$ nor that $\mathsf{F}^*\,B$.

We identify the basic (*0-depth*) logic of our hierarchy of approximations with the inferences that an agent can draw without making hypotheses about the 'objective' informational situation concerning the whole of Ω. In other words, without making hypothetical assumptions that go beyond the information that he holds. We shall show that a natural proof-theoretic characterization of this basic logic is obtained by means of the set of introduction and elimination (*intelim*) rules respectively displayed in Tables 2 and 3. The analogous 0-depth system for **CPL** in [30, 31] is characterized by the intelim rules obtained by removing all the starred signs, replacing them with the unstarred signs T and F, interpreted as 'only true' and 'only false', and eliminating duplicates. Note that the characterization of the basic logic bears some resemblance with natural deduction, but does not have discharge rules, since no hypothetical reasoning is involved. Besides, observe that the intelim rules for disjunction and conjunction are dual of each other, and that a sentence and its negation are treated in a symmetric way. Moreover, in the elimination rules, we shall refer to the premise containing the connective that is to be eliminated as *major* and to the other premise as *minor*. In turn, given that the intelim rules have all a linear format, their application generates *intelim sequences*. Namely, finite sequences $(\varphi_1, \ldots, \varphi_n)$ of S-formulae such that, for every $i = 0, \ldots, n$, either φ_i is an assumption or it is the conclusion of the application of an intelim rule to preceding S-formulae. In Figure 1 we show a simple example of an intelim sequence, where each assumption is marked with an '@'.

The intelim rules are all sound, but not complete for full **FDE**. Indeed, as we shall show below, these rules just characterize the basic logic in the hierarchy of approximations. Completeness for full **FDE** is obtained by adding only two branching structural rules, according to which we are allowed to: (i) append both $\mathsf{T}\,A$ and $\mathsf{F}\,A$ as sibling nodes to the last element of any intelim sequence; (ii) append both $\mathsf{T}^*\,A$ and $\mathsf{F}^*\,A$ in a similar way. The intuitive meaning of these rules is that one of the two cases must obtain considering the whole of Ω even if the agent has no actual information about which is the case. In this sense, we call the information expressed by each conjugate S-formula *virtual* information; i.e. hypothetical information that the agent does not hold, but she temporarily assumes as if she held it. We respectively call these branching rules *PB* and *PB** as they are closely

TABLE 3. Elimination rules for the standard **FDE** connectives

$$
\frac{\begin{array}{c}\mathsf{F}\,A\wedge B\\ \mathsf{T}\,A\end{array}}{\mathsf{F}\,B}
\qquad
\frac{\begin{array}{c}\mathsf{F}\,A\wedge B\\ \mathsf{T}\,B\end{array}}{\mathsf{F}\,A}
\qquad
\frac{\begin{array}{c}\mathsf{F}^*\,A\wedge B\\ \mathsf{T}^*\,A\end{array}}{\mathsf{F}^*\,B}
\qquad
\frac{\begin{array}{c}\mathsf{F}^*\,A\wedge B\\ \mathsf{T}^*\,B\end{array}}{\mathsf{F}^*\,A}
$$

$$
\frac{\mathsf{T}\,A\wedge B}{\mathsf{T}\,A}
\qquad
\frac{\mathsf{T}\,A\wedge B}{\mathsf{T}\,B}
\qquad
\frac{\mathsf{T}^*\,A\wedge B}{\mathsf{T}^*\,A}
\qquad
\frac{\mathsf{T}^*\,A\wedge B}{\mathsf{T}^*\,B}
$$

$$
\frac{\begin{array}{c}\mathsf{T}\,A\vee B\\ \mathsf{F}\,A\end{array}}{\mathsf{T}\,B}
\qquad
\frac{\begin{array}{c}\mathsf{T}\,A\vee B\\ \mathsf{F}\,B\end{array}}{\mathsf{T}\,A}
\qquad
\frac{\begin{array}{c}\mathsf{T}^*\,A\vee B\\ \mathsf{F}^*\,A\end{array}}{\mathsf{T}^*\,B}
\qquad
\frac{\begin{array}{c}\mathsf{T}^*\,A\vee B\\ \mathsf{F}^*\,B\end{array}}{\mathsf{T}^*\,A}
$$

$$
\frac{\mathsf{F}\,A\vee B}{\mathsf{F}\,A}
\qquad
\frac{\mathsf{F}\,A\vee B}{\mathsf{F}\,B}
\qquad
\frac{\mathsf{F}^*\,A\vee B}{\mathsf{F}^*\,A}
\qquad
\frac{\mathsf{F}^*\,A\vee B}{\mathsf{F}^*\,B}
$$

$$
\frac{\mathsf{T}\,\neg A}{\mathsf{F}^*\,A}
\qquad
\frac{\mathsf{F}\,\neg A}{\mathsf{T}^*\,A}
\qquad
\frac{\mathsf{T}^*\,\neg A}{\mathsf{F}\,A}
\qquad
\frac{\mathsf{F}^*\,\neg A}{\mathsf{T}\,A}
$$

$$
\begin{array}{l}
\mathsf{T}\,\neg(A\vee B)^{@}\\
\mathsf{T}\,\neg C^{@}\\
\mathsf{F}^*\,A\vee B\\
\mathsf{F}^*\,A\\
\mathsf{F}^*\,C\\
\mathsf{F}^*\,A\vee C\\
\mathsf{T}\,\neg(A\vee C)
\end{array}
$$

FIGURE 1. An itelim sequence

related to a *generalized* Principle of Bivalence:[6]

$$
\frac{}{\mathsf{T}\,A \mid \mathsf{F}\,A}
\qquad\qquad
\frac{}{\mathsf{T}^*\,A \mid \mathsf{F}^*\,A}
$$

For **CPL** only the first rule, with T and F interpreted as 'only true' and 'only false', makes sense and is sufficient for completeness. Given that the sets of starred and unstarred rules look identical, one might wonder why we need both sets and ultimately four signs. The reason is given by the negation connective. For example, according to the **FDE**-table for $\neg$ and the intended interpretation of the signed formulae, $\mathsf{T}\,\neg A$ ($v(\neg A) \in \{\mathbf{t}, \mathbf{b}\}$) implies $\mathsf{F}^*\,A$ ($v(A) \in \{\mathbf{f}, \mathbf{b}\}$), and not $\mathsf{F}\,A$ as one would expect classically. The other cases involving negation are, of course, analogous [see 46].

With the addition of *PB* and *PB** to the stock of rules, deductions are represented by downward-growing trees, which brings the method somewhat closer to tableaux. Each application of *PB* or *PB** stands for the introduction of virtual information about the imprecise value of a formula A, which

[6]Generalizations of the rule of bivalence have been fruitfully used in the context of many-valued and substructural logics [see 20, 33, 49].

we shall respectively call the *PB-formula* or *PB*-formula*. Note once again that, whereas signed formulae of the form $\mathsf{T}\,A$ and $\mathsf{F}^*\,A$ represent information that may be empirically obtained (when A turns out to be **b**), signed formulae of the form $\mathsf{T}^*\,A$ and $\mathsf{F}\,A$ are obtainable only by applying *PB* or *PB**. In turn, any S-formulae appended via those branching rules will be called a *virtual assumption*. Now, *PB* and *PB** are essentially cut rules that may introduce formulae of arbitrary degree. However, as we will show in Lemma 5.4, their application can be restricted so as to satisfy the subformula property. Moreover, from our informational viewpoint, the main conceptual advantage of this proof-theoretic characterization consists in that it clearly separates the *intelim rules* that fix the meaning of the connectives in terms of the information that an agent holds from the two *structural rules* that introduce virtual information (*PB* and *PB**).

Intuitively, the more virtual information needs to be invoked via *PB* or *PB** the harder the inference is for the agent, both from the computational and the cognitive viewpoint. In this sense, the nested applications of *PB* and *PB** provide a sensible measure of inferential *depth*. This naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to **FDE** in terms of the maximum number of nested applications of *PB* and *PB** that are allowed. In turn, such a hierarchy can be intuitively associated with a hierarchy of increasingly idealized agents with more and more—albeit always bounded—cognitive and computational resources or inferential power. Note, however, that the inferential depth associated with an agent is not intended to be interpreted as an upper bound on her inferential power. Rather, it is understood as the maximum depth for which it is guaranteed that, if she possesses the information explicitly carried by the assumptions, she possesses the information explicitly carried by the conclusion.

Before giving definitions and results, we remark that (i) unlike the branching rules of Smullyan-style tableaux, our branching rules are *structural* in that they do not involve any specific logical operator; (ii) the elimination rules, together with the branching rules, were early introduced in [26] as constituting a refutation method for full **FDE** called RE_{fde}. So, the completeness of RE_{fde} trivially implies the completeness of the system presented in this paper. However, our intelim method can be used as a direct-proof method as well as a refutation method, and leads to more powerful approximations. A direct completeness proof can also be given based on the semantics, which yields the subformula property. In this paper we choose to prove a more general version of the subformula property by means of proof transformations.

DEFINITIONS 6

- Let $X = \{\varphi_1, \ldots, \varphi_m\}$. Then $\mathcal{T}$ is an *intelim tree for X* if there is a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n)$ such that $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $(\varphi_1, \ldots, \varphi_m)$, $\mathcal{T}_n = \mathcal{T}$, and for each $i < n$, $\mathcal{T}_{i+1}$ results from $\mathcal{T}_i$ by an application of an intelim rule to preceding S-formulae in the same branch, or by an application of *PB* or *PB**.
- A branch of an intelim tree is *closed* if it contains an S-formula φ and its conjugate $\bar{\varphi}$; otherwise, it is *open*.
- An intelim tree is said to be *closed* when all its branches are closed; otherwise, it is *open*.
- An *intelim proof of φ from X* is an intelim tree $\mathcal{T}$ for X such that φ occurs in all open branches of $\mathcal{T}$.
- An *intelim refutation of X* is a closed intelim tree $\mathcal{T}$ for X.

Note that every refutation of X is, simultaneously, a proof of φ from X, for every φ. This because there are no open branches and so the condition that φ occurs at the end of all open branches is vacuously satisfied. This is, of course, a kind of explosivity, but it regards *signed* formulae, and it

$$\begin{array}{l} \mathsf{T}\,(A \vee B) \wedge \neg A^{@} \\ \mathsf{F}\,B^{@} \\ \mathsf{T}\,A \vee B \\ \mathsf{T}\,\neg A \\ \mathsf{T}\,A \\ \mathsf{F}^{*}\,A \end{array}$$

FIGURE 2. A single-branch open itelim tree

is compatible with the non-explosivity regarding formulae in **FDE**. The reason of that compatibility is that a set consisting of S-formulae all of the form $\mathsf{T}A$ cannot lead to explosion because there cannot be an intelim refutation of such a set. To begin with, starting from a set $\mathsf{T}\,\Gamma$, there is no way of obtaining S-formulae of the form $\mathsf{F}A$ or T^*A by applying only intelim rules. Starting from a set $\mathsf{T}\,\Gamma$, the only way of obtaining formulae of such forms is by applying *PB* or *PB** and, thus, adding virtual information. Nonetheless, a set $\mathsf{T}\,\Gamma$ cannot lead to explosion even if we add virtual information when unfolding the information contained in $\mathsf{T}\,\Gamma$. In fact, as the following result shows, for a set of S-formulae X to lead to explosion it must contain (in itself) S-formulae of the form $\mathsf{F}A$ or T^*A, i.e. virtual information.

PROPOSITION 4.1
Any intelim tree for a set $\mathsf{T}\,\Gamma$ has at least a branch containing only S-formulae of the form $\mathsf{T}A$ or F^*A.

PROOF. By an easy induction on the number of nodes of the intelim tree. □

The above proposition implies that any intelim tree for a set $\mathsf{T}\,\Gamma$ is open and, thus, that there is no refutation of such a set. This fact regarding our proof-theoretic characterization of **FDE** corresponds to the fact regarding its 4-valued semantics according to which all the elements of any set of formulae can have a designated truth-value. More specifically, it corresponds to the fact that for any formula A there is a **4**-valuation v such that $v(A) = \mathbf{b}$; namely, v such that for all $p \in At(\mathcal{L}), v(p) = \mathbf{b}$. Note that a set $\mathsf{T}\,\Gamma$ may well contain $\mathsf{T}A$ and $\mathsf{T}\neg A$, or either of them may well be obtained from that set by applying the rules, which precisely amounts to the formula A having the truth-value **b**. However, that pair of S-formulae does not close a branch.

Moreover, as it is usual in refutation methods, our intelim method can be used to obtain counterexamples from open branches [see also 46]. For instance, in Figure 2, we can extract counterexamples for disjunctive syllogism out of the open branch of the tree as follows. The branch contains $\mathsf{F}B$, so B can be either **f** or **n**. It also contains both $\mathsf{T}A$ and F^*A, so A should be one of **t** or **b**, or A should be one of **f** or **b**. Then, A should be exactly **b**. We thus obtain two counterexamples for disjunctive syllogism. Namely, a **4**-valuation v such that $v(A) = \mathbf{b}$ and $v(B) = \mathbf{f}$, and another **4**-valuation v such that $v(A) = \mathbf{b}$ and $v(B) = \mathbf{n}$.

Now, as mentioned above, *PB* and *PB** may introduce formulae of arbitrary degree. However, the set of formulae that can be used as *PB*-formulae or *PB**-formulae can be bounded in a variety of ways without loss of completeness. We call this set *virtual space* and define it as a function f of the set $\Gamma \cup \{A\}$, consisting of the premises Γ and of the conclusion A of the given inference. The strictest way of bounding the virtual space consists in allowing as *PB*-formulae and *PB**-formulae only atomic formulae that occur in $\Gamma \cup \{A\}$. A more liberal option is allowing only subformulae of the formulae in $\Gamma \cup \{A\}$. More generally, let $\mathcal{F}$ be the set of all functions f on the finite subsets of $F(\mathcal{L})$ such that:[7] (i) for all Δ, $\mathsf{at}(\Delta) \subseteq f(\Delta)$; (ii) $f(\Delta)$ is closed under subformulae, i.e. $\mathsf{sub}(f(\Delta)) = f(\Delta)$; (iii) the *size* of $f(\Delta)$ is bounded above by a polynomial in the size of Δ, i.e. $|f(\Delta)| \leq p(|\Delta|)$ for some fixed polynomial p. (This last requirement will be essential in order to define *tractable* approximations below.) The choice of an specific function to yield suitable values of the virtual space for each particular deduction problem is the result of decisions that are conveniently made by the system designer, depending on the intended application.[8] In turn, the functions in $\mathcal{F}$ are partially ordered by the relation $\trianglelefteq$ such that $f_1 \trianglelefteq f_2$ iff, for every finite $\Delta, f_1(\Delta) \subseteq f_2(\Delta)$.

Distinguished examples of functions in $\mathcal{F}$ are the identity function $f(\Delta) = \Delta$, sub and at. However, in general, $f(\Delta)$ may contain formulae that are not in $\mathsf{sub}(\Delta)$. For instance, the operation f that maps Δ to the set of all formulae of bounded degree that can be built out of $\mathsf{sub}(\Delta)$ and $\mathsf{at}(\Delta)$ is also in $\mathcal{F}$. Thus, our intelim method allows for (possibly shorter) deductions that do not have the subformula property simply by permitting applications of *PB* or *PB** to formulae that are not subformulae either of the premises or of the conclusion. However, even in this latter deductions the virtual space is still bounded.

The branching rules are not the unique rules of our intelim method that may bring about violations of the subformula property. The introduction rules could in principle be indefinitely applied, leading to ever more complex formulae. Nonetheless, as we shall show below, the application of both kind of rules can be restricted so as to satisfy the subformula property. More specifically, we shall show that every intelim proof of φ from X (intelim refutation of X) can be transformed into an intelim proof of φ from X (an intelim refutation of X) with the subformula property.

5 Subformula property

The subformula property (SFP, for short) is a key property of logical systems in that it allows us to search for proofs or refutations by *analytic methods*; i.e. by considering solely deduction steps involving formulae that are 'contained' in the assumptions, or also in the conclusion in the case of proofs. This implies a drastic reduction of the search space which is crucial for the purpose of automated deduction. When it comes to propositional logics, this search space is finite for each putative inference, paving the way for decision procedures. Particularly, in our intelim method, the SFP guarantees that we can impose a bound on the applications of *PB* and *PB**, that could in principle be applied to arbitrary formulae, with no loss of deductive power. Similarly, it guarantees that we can impose a bound on the sensible applications of introduction rules, which could in principle be indefinitely applied, yielding ever more complex formulae.

DEFINITION 5.1
An intelim proof $\mathcal{T}$ of φ from X (an intelim refutation of X) has the *subformula property* if, for every S-formula ψ occurring in $\mathcal{T}$, $\psi^u \in \mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\psi^u \in \mathsf{sub}(X^u)$).

Now, consider the intelim sequences of Figure 3. The first one is a proof of $\mathsf{T}^* \neg s$ from $\{\mathsf{T}^* \neg p, \mathsf{T}\, q, \mathsf{T}^* \neg r \vee \neg s, \mathsf{T}\, r\}$. The second one is a proof of (an arbitrary) $\mathsf{T}\, q$ from $\{\mathsf{T}\, p, \mathsf{F}\, p\}$; i.e. an instance of the explosivity of our intelim method. Note that both proofs are *redundant*. In the first

[7] The *size of a formula A*, denoted by $|A|$, is the total number of occurrences of symbols in A; whereas the *size of a finite set of formulae* Γ, denoted by $|\Gamma|$, is defined as $\sum_{A \in \Gamma} |A|$.

[8] In the case of the approximations defined below, such decisions affect the deductive power of each given approximation, and so the 'speed' at which the approximation process converges to the limiting logic at issue.

$\mathsf{T}^* \neg p^{@}$	$\mathsf{T}\, p^{@}$
$\mathsf{T}\, q^{@}$	$\mathsf{F}\, p^{@}$
$\mathsf{T}^* \neg r \vee \neg s^{@}$	$\mathsf{T}\, p \vee q$
$\mathsf{T}\, r^{@}$	$\mathsf{T}\, q$
$\mathsf{T}\, r \vee p$	
$\mathsf{F}\, p$	
$\mathsf{T}\, r$	
$\mathsf{F}^* \neg r$	
$\mathsf{T}^* \neg s$	

FIGURE 3. Redundant itelim sequences

proof, the S-formula $\mathsf{T}\, r \vee p$ is first introduced (from premise $\mathsf{T}\, r$) and then eliminated (using the minor premise $\mathsf{F}\, p$) to re-obtain the S-formula $\mathsf{T}\, r$ which was already contained in the sequence; i.e. this proof contains circular reasoning. In the second proof, the S-formula $\mathsf{T}\, p \vee q$ is first introduced (from premise $\mathsf{T}\, p$) and then eliminated (using $\mathsf{F}\, p$ as minor premise); yet, the sequence was already closed before the application of the disjunction introduction and so, by Definitions 6, the closed sequence $\mathsf{T}\, p, \mathsf{F}\, p$ was already a proof of $\mathsf{T}\, q$ from $\mathsf{T}\, p$ and $\mathsf{F}\, p$.

The same kind of redundancy is present whenever a formula is, simultaneously, the conclusion of an introduction and the major premise of an elimination.

DEFINITIONS 7

An occurrence of an S-formula φ in an intelim tree $\mathcal{T}$ is: (i) a *detour* if φ is both the conclusion of an introduction and the major premise of an elimination; (ii) *idle* if it is not the terminal node of its branch, it is not used as premise of some application of an intelim rule, and it is not the conjugate of some S-formula occurring in the same branch.

DEFINITIONS 8

Given an intelim tree $\mathcal{T}$, a *path* in $\mathcal{T}$ is a finite sequence of nodes such that the first node is the root of $\mathcal{T}$ and each of the subsequent nodes is an immediate successor of the previous one. A path is *closed* if it contains both φ and $\bar{\varphi}$ for some φ.

Note that, according to the above definition, every branch is a maximal path.

DEFINITION 5.2

Let $\mathcal{T}$ be an intelim proof of φ from X (an intelim refutation of X). $\mathcal{T}$ is *non-redundant* if it satisfies the following conditions:

1. it contains no idle occurrences of S-formulae;
2. none of its branches contains more than one occurrence of the same S-formula;
3. none of its branches properly includes a closed path.

Observe that if an intelim proof or refutation contains a detour, then either condition 2 or 3 above is violated. Thus:

LEMMA 5.3
If an intelim proof or refutation $\mathcal{T}$ is non-redundant, then it contains no detours.

PROOF. By the definitions above and inspection of the intelim rules. Every detour makes the tree redundant. □

Now, turning an intelim proof or refutation $\mathcal{T}$ into a non-redundant one (with no increase in the size of the proof or refutation) is computationally easy, in that it only involves the following pruning steps:

1. check if there are closed paths and remove whatever follows after them;
2. remove any repetition of S-formulae in the same branch;
3. check if there are idle occurrences of S-formulae, and
4. for each idle occurrence of an S-formula φ:
   - if φ is the conclusion of an application of an intelim rule, just remove φ from $\mathcal{T}$;
   - if φ is a virtual assumption introduced by an application of *PB* or *PB**, remove both φ and the whole subtree generated by its conjugate S-formula $\bar{\varphi}$ introduced in the same application of *PB* or *PB**; then attach the subtree below φ to the immediate predecessor of φ.

It is easy to verify that the result of this procedure is still an intelim proof of the same conclusion from the same premises, or an intelim refutation of the same assumptions.

In turn, given a proof $\mathcal{T}$ of φ from X (a refutation of X), and any operation $f \in \mathcal{F}$, we say that an application of *PB* or *PB** in $\mathcal{T}$ is *f-analytic* if its *PB*-formula or *PB**-formula is in $f(X^u \cup \{\varphi^u\})$ ($f(X^u)$); i.e. in the virtual space defined by the function f. Recall that the latter is, by definition, closed under subformulae and polynomially bounded. When $f = \mathsf{sub}$, i.e. the virtual space consists exactly of the subformulae of $X^u \cup \{\varphi^u\}$ (X^u), we just say that the application of *PB* or *PB** is *analytic*. Thus:

LEMMA 5.4
Given any $f \in \mathcal{F}$, every intelim proof $\mathcal{T}$ of φ from X (intelim refutation of X) can be transformed into an intelim proof of $\mathcal{T}'$ φ from X (intelim refutation $\mathcal{T}'$ of X) such that every application of *PB* and *PB** in $\mathcal{T}$ is f-analytic.

PROOF. We use the notation $\begin{array}{c}\mathcal{T}\\ \psi\end{array}$ to denote either an empty intelim tree or a non-empty intelim tree such that ψ is one of its terminal nodes. The proof is by lexicographic induction on $\langle \gamma(\mathcal{T}), \kappa(\mathcal{T})\rangle$, where $\gamma(\mathcal{T})$ denotes the maximum degree of a *PB*-formula or a *PB**-formula in $\mathcal{T}$ that is not f-analytic, and $\kappa(\mathcal{T})$ denotes the number of occurrences of such non-f-analytic *PB*-formulae or *PB**-formulae of maximal degree. Let $\gamma(\mathcal{T}) = m > 0$ and let A be a *PB*-formula or a *PB**-formula of degree m. There are several cases depending on the logical form of A and on whether A is *PB*-formula or a *PB**-formula. We sketch only the case where $A = B \wedge C$ and A is a *PB**-formula; the other cases being similar. So, $\mathcal{T}$ has the following form:

$$\begin{array}{c} \mathcal{T}_a \\ \psi \\ \begin{array}{cc} \mathsf{T}^* B \wedge C & \mathsf{F}^* B \wedge C \\ \mathcal{T}_b & \mathcal{T}_c \end{array} \end{array}$$

where $\mathcal{T}_b$ and $\mathcal{T}_c$ are intelim trees such that each of their open branches contains φ, or are both closed intelim trees. Let $\mathcal{T}'$ be the following intelim tree:

$$
\begin{array}{c}
\mathcal{T}_a\\
\psi\\
\begin{array}{cc}
\begin{array}{c}
\mathsf{T}^* B\\
\begin{array}{cc}
\begin{array}{c}\mathsf{T}^* C\\ \mathsf{T}^* B \wedge C\\ \mathcal{T}_b\end{array} &
\begin{array}{c}\mathsf{F}^* C\\ \mathsf{F}^* B \wedge C\\ \mathcal{T}_c\end{array}
\end{array}
\end{array} &
\begin{array}{c}\mathsf{F}^* B\\ \mathsf{F}^* B \wedge C\\ \mathcal{T}_c\end{array}
\end{array}
\end{array}
$$

Clearly, $\mathcal{T}'$ is an intelim proof of φ from X (an intelim refutation of X). Moreover, either $\gamma(\mathcal{T}') < \gamma(\mathcal{T})$, or $\gamma(\mathcal{T}') = \gamma(\mathcal{T})$ and $\kappa(\mathcal{T}') < \kappa(\mathcal{T})$. □

In fact, the transformations used in the proof of the above lemma show that every intelim tree can be turned into an equivalent one in which all the *PB*-formulae and *PB**-formulae are atomic. Thus, in principle, we could reformulate the notion of intelim tree in such a way that *PB* and *PB** are applied only to atomic formulae without loss of completeness. Nevertheless, if we demand that the applications of *PB* and *PB** be restricted to atomic formulae, the property of being an intelim tree is no longer preserved under uniform substitutions of the atomic formulae occurring in the tree with arbitrary formulae.[9] On the other hand, if we require that the notion of intelim tree be restricted so as to permit only analytic applications of *PB* and *PB** (i.e. f-analytic applications with $f = \mathsf{sub}$), the property of being an intelim tree is indeed invariant under uniform substitutions.

The following theorem states the SFP of our intelim method when $f = \mathsf{sub}$:[10]

THEOREM 5.5 (Generalized SFP).
For every $f \in \mathcal{F}$, if $\mathcal{T}$ is an intelim proof of φ from X (an intelim refutation of X) such that (i) $\mathcal{T}$ is non-redundant, and (ii) every application of *PB* and *PB** in $\mathcal{T}$ is f-analytic, then for every S-formula ψ occurring in $\mathcal{T}$,

$$\psi^u \in f(X^u \cup \{\varphi^u\}) \cup \mathsf{sub}(X^u \cup \{\varphi^u\})$$

if $\mathcal{T}$ is a proof of φ from X, or

$$\psi^u \in f(X^u) \cup \mathsf{sub}(X^u)$$

if $\mathcal{T}$ is a refutation of X.

PROOF. Let $\mathcal{T}$ be a intelim proof of φ from X (refutation of X) satisfying (i) and (ii), and suppose that there are S-formulae ω in $\mathcal{T}$ such that $\omega^u \notin f(X^u \cup \{\varphi^u\}) \cup \mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\omega^u \notin f(X^u) \cup \mathsf{sub}(X^u)$). Let us call such S-formulae *spurious*. Let ψ be a spurious formula such that ψ^u is of maximal degree in $\mathcal{T}$. Then ψ cannot result from the application of an elimination rule, otherwise $\mathcal{T}$ would contain an spurious formula whose unsigned part is of strictly greater degree; namely, the major premise of this elimination. Moreover, given that $\mathcal{T}$ contains only f-analytic applications of *PB* and *PB**

[9] Moreover, when we shall define the notion of *depth* of an intelim tree below, it will be apparent that each application of the transformations used in the proof of the lemma increases the depth of the tree. So, it is convenient to use them only to the extent in which it is needed to remove applications of *PB* or *PB** which are not f-analytic.

[10] Note that whenever $\Delta \subseteq f(\Delta)$, then also $\mathsf{sub}(\Delta) \subseteq f(\Delta)$.

according to (ii), no spurious S-formula can occur in it as a virtual assumption introduced by an application of *PB* or *PB**. Therefore, ψ must be the conclusion of an introduction. Since, according to (i), $\mathcal{T}$ is non-redundant, it contains no idle occurrences of S-formulae and so either (a) $\psi = \bar{\theta}$ for some θ occurring in the same branch or (b) ψ is used as a premise of a rule application. However, both cases are impossible. Regarding (a), by the same arguments just used for ψ, θ (the conjugate of ψ) can only be the conclusion of an introduction. Then, it is not difficult to see, by inspection of the introduction rules, that case (a) implies that one of the premises of the introduction of $\psi = \bar{\theta}$ must be the conjugate of one of the premises of the introduction of θ. So, one of the branches of $\mathcal{T}$ properly contains a closed path, against the assumption that $\mathcal{T}$ is non-redundant. As for (b), first note that ψ cannot be the minor premise of an elimination, otherwise there would be again an spurious formula whose unsigned part is of greater degree in $\mathcal{T}$; namely, the major premise of this elimination. Moreover, ψ cannot be used in $\mathcal{T}$ as major premise of an elimination, otherwise ψ would be a detour and, by Lemma 5.3, $\mathcal{T}$ would be redundant, against hypothesis (i). □

6 Depth-bounded approximations to FDE

DEFINITIONS 9
The *depth* of an intelim tree $\mathcal{T}$ is the maximum number of virtual assumptions occurring in a branch of $\mathcal{T}$. An intelim tree $\mathcal{T}$ is a *k-depth intelim proof of φ from X* (a *k-depth intelim refutation of X*) if $\mathcal{T}$ is an intelim proof of φ from X (an intelim refutation of X) and $\mathcal{T}$ is of depth k.

Note that a 0-depth intelim tree is nothing but an intelim sequence. Examples of, respectively, two proofs of depth 1, a refutation of depth 1 and a proof of depth 2, all with the SFP, are given in Figure 4. Again, each assumption is marked with an '@'.

DEFINITIONS 10
For all X, φ,

- φ is *0-depth deducible* from X, $X \vdash_0 \varphi$, iff there is a 0-depth intelim proof of φ from X;
- X is *0-depth refutable*, $X \vdash_0$, iff there is a 0-depth intelim refutation of X.

NOTATION 11
We shall abuse of the same relation symbol '$\vdash_0$' to denote 0-depth deducibility and refutability.

PROPOSITION 6.1
$\langle \mathcal{L}, \vdash_0 \rangle$ is a (finitary) Tarskian propositional logic; i.e. $\vdash_0$ satisfies reflexivity, monotonicity, cut and structurality.

PROOF. By the definitions above. □

Now, the following definition will be useful for showing the tractability of our approximations below:

DEFINITION 6.2
A set X is *intelim saturated* if it satisfies the following conditions:

1. for no A, $\mathsf{T}\,A$ and $\mathsf{F}\,A$, nor $\mathsf{T}^*\,A$ and $\mathsf{F}^*\,A$, are both in X;

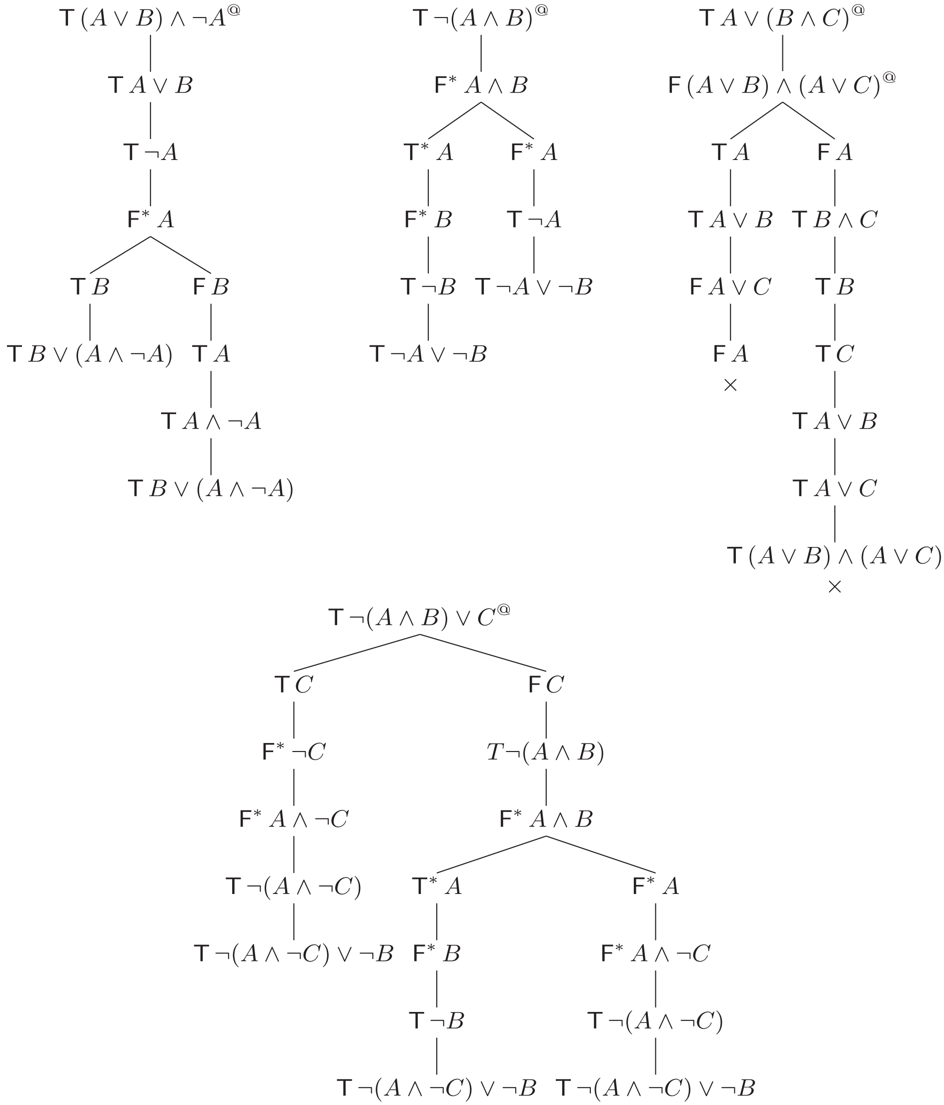

FIGURE 4. *k*-depth intelim proofs and refutations

2. for every φ, if φ follows from some subset of X by an application of an intelim rule, then $\varphi \in X$.

In turn, the notion of *k*-depth deducibility depends not only on the depth at which the use of virtual information is recursively allowed, but also on the virtual space discussed above. So, finally:

DEFINITIONS 11
For all X, φ and for all $f \in \mathcal{F}$,

- $X \vdash_0^f \varphi$ iff $X \vdash_0 \varphi$;
- for $k > 0$, $X \vdash_k^f \varphi$ iff $X \cup \{\psi\} \vdash_{k-1}^f \varphi$ and $X \cup \{\bar{\psi}\} \vdash_{k-1}^f \varphi$ for some $\psi^u \in f(X^u \cup \{\varphi^u\})$.

When $X \vdash_k^f \varphi$, we say that φ is *deducible at depth k from X over the f-bounded virtual space*. The above definition covers also the case of k-depth refutability by assuming $X \vdash_k^f$ as equivalent to $X \vdash_k^f \varphi$ for all φ. When $X \vdash_k^f$, we say that X *is refutable at depth k over the f-bounded virtual space*.

NOTATION 12
We shall abuse of the same relation symbol '$\vdash_k^f$' to denote k-depth deducibility and refutability over the f-bounded virtual space.

Observe that in the above definitions the pair of S-formulae, ψ and $\bar{\psi}$, denote a pair of (conjugate) virtual assumptions introduced by respectively *PB* or *PB**. Thus, according to the definitions, $X \vdash_k^f \varphi$ iff the conclusion φ is obtained at depth $k-1$ by introducing *either* $\mathsf{T}\,A$ and $\mathsf{F}\,A$, *or* $\mathsf{T}^*\,A$ and $\mathsf{F}^*\,A$, as virtual assumptions—for some A in the virtual space defined by f. This corresponds to the fact that, in our intelim method, a formula φ may be obtained *at a certain depth* by introducing whichever $\mathsf{T}\,A$ or $\mathsf{F}\,A$ by an application of *PB* but not by introducing $\mathsf{T}^*\,A$ or $\mathsf{F}^*\,A$ by an application of *PB** and vice versa.

Now, it follows immediately from Definitions 9 and 11 that:

PROPOSITION 6.3
For all X, φ and all $f \in \mathcal{F}$, $X \vdash_k^f \varphi$ ($X \vdash_k^f$) iff there is a k-depth intelim proof of φ from X (a k-depth intelim refutation of X) such that all its *PB*-formulae and *PB**-formulae are in $f(X^u \cup \{\varphi^u\})$ ($f(X^u)$).

PROPOSITION 6.4
The k-depth deducibility relations $\vdash_k^f$ satisfy reflexivity, monotonicity, but not cut.

PROOF. By the definitions above. □

However, it is easy to verify that the relations $\vdash_k^f$ satisfy the following version of cut:

Depth-bounded cut: If $X \vdash_j^f \varphi$ and $X \cup \{\varphi\} \vdash_k^f \psi$, then $X \vdash_{j+k}^f \psi$.

Moreover, the relations $\vdash_k^f$ may not be structural in that structurality depends on the function f that defines the virtual space. For example, $\vdash_k^{\mathsf{sub}}$ is structural, while $\vdash_k^{\mathsf{at}}$ is not. In general, structurality can be imposed by restricting the operations in $\mathcal{F}$ to those such that, for all substitution σ and all Δ, $\sigma(f(\Delta)) \subseteq f(\sigma(\Delta))$. This is not satisfied if $f = \mathsf{at}$, but it is satisfied if $f(\Delta) = \mathsf{sub}(\Delta)$, or $f(\Delta)$ is the set of all formulae of given bounded degree that can be built out of $\mathsf{sub}(\Delta)$. Further, since $\vdash_0$ is monotonic, its successors are ordered: $\vdash_j^f \subseteq \vdash_k^f$ whenever $j \leq k$. The transition from $\vdash_k^f$ to $\vdash_{k+1}^f$ corresponds to an increase in the depth at which the nested use of virtual information (restricted to formulae in the virtual space defined by f) is allowed. Note also that $\vdash_j^{f_1} \subseteq \vdash_k^{f_2}$ whenever $f_1 \trianglelefteq f_2$.

6.1 Tractability

We now show that the decision problem for the k-depth logics is tractable. Theorem 5.5 immediately suggests a decision procedure for k-depth deducibility: to establish whether φ is k-depth deducible from a finite set X we apply the intelim rules, together with *PB* and *PB** up to a number k of times, in all possible ways starting from X and restricting to applications which preserve the SFP. If the resulting intelim tree is closed or φ occurs at the end of all its open branches, then φ is k-depth deducible from X, otherwise it is not. We shall first show the tractability of the 0-depth logic, and then the tractability of the k-depth logics, $k > 0$.

DEFINITION 6.5
The *subformula graph* for X^u, G, is the oriented graph $\langle V, E\rangle$ such that $V = \mathsf{sub}(X^u)$ and $\langle A, B\rangle \in E$ iff A is an immediate subformula of B.

DEFINITION 6.6
A *module* is a set consisting of a non-atomic formula, called the *top formula* of the module, and of its immediate subformulae, called the *secondary formulae* of the module.

DEFINITION 6.7
A *G-module* is any subgraph M of G whose set of vertices is a module; i.e., consists of a formula with all its immediate subformulae. The *top formula* of M is the top formula of the underlying module.

DEFINITION 6.8
A *labelled subformula graph* for X^u is a pair $\langle G, \lambda\rangle$, where G is the subformula graph for X^u, and λ is a relation $\lambda \subseteq V \times \{\mathsf{T}, \mathsf{F}, \mathsf{T}^*, \mathsf{F}^*\}$, called the *labelling relation* and such that:

i. each $A \in V$ is related to at most two elements in $\{\mathsf{T}, \mathsf{F}, \mathsf{T}^*, \mathsf{F}^*\}$;
ii. for no $A \in V$, $\langle A, \mathsf{T}\rangle$ and $\langle A, \mathsf{F}\rangle$ are both in λ;
iii. for no $A \in V$, $\langle A, \mathsf{T}^*\rangle$ and $\langle A, \mathsf{F}^*\rangle$ are both in λ.

DEFINITIONS 12
An *intelim graph for X^u based on Y^u*, with $Y^u \subseteq X^u$, is a labelled subformula graph $\langle G, \lambda\rangle$ for X^u satisfying the following conditions:

1. for every $\mathsf{S}A \in Y$, $\mathsf{S} \in \lambda(A)$;
2. for every $A \in \mathsf{sub}(X^u)$ such that $A \notin Y^u$, $\lambda(A)$ is defined and equal to $\{\mathsf{T}\}$, $\{\mathsf{F}\}$, $\{\mathsf{T}^*\}$, $\{\mathsf{F}^*\}$, $\{\mathsf{T}, \mathsf{T}^*\}$, $\{\mathsf{T}, \mathsf{F}^*\}$, $\{\mathsf{F}, \mathsf{T}^*\}$, or $\{\mathsf{F}, \mathsf{F}^*\}$ *only if* there are $B_1, \ldots, B_k \in \mathsf{sub}(X^u)$ such that $\mathsf{T}A$, $\mathsf{F}A$, T^*A, or F^*A follows from

$$\{\mathsf{T}B_i \mid 0 \le i \le k, \mathsf{T} \in \lambda(B_i)\} \cup \{\mathsf{F}B_i \mid 0 \le i \le k, \mathsf{F} \in \lambda(B_i)\}$$
$$\cup \{\mathsf{T}^*B_i \mid 0 \le i \le k, \mathsf{T}^* \in \lambda(B_i)\} \cup \{\mathsf{F}^*B_i \mid 0 \le i \le k, \mathsf{F}^* \in \lambda(B_i)\}$$

by an application of an intelim rule.[11]

$\lambda(A) = \emptyset$ whenever the image of A under λ is undefined, and we say that the *initial intelim graph for X^u based on Y^u* is the intelim graph for X^u based on Y^u such that $\lambda(A) = \emptyset$ for all $A \notin Y^u$. An intelim graph for X^u is *completed* if it satisfies also the converse of 2.

Now, let the symbol '$\top$' stand for the 'inconsistent labeling relation'. This is only a way of speaking to mean that there is no labelling relation consistent with the intelim rules. Simple decision

[11] Note that this procedure applies to arbitrary rules of the form $\varphi_1, \ldots, \varphi_k/\varphi_{k+1}$. So, adding the subformula rule specified in 2 would not increase the complexity of the decision procedure, and may reduce the depth of the inference.

procedures for 0-depth refutability and deducibility are illustrated in Algorithms 1 and 2, and consist in building the initial intelim graph for the set of formulae that are mentioned in the specification of the problem—i.e. just assumptions for refutation problems, and assumptions plus the conclusion for deduction problems—and turning it into a completed intelim graph in accordance with the intelim rules. Both algorithms call the subroutine `Expand` described in Algorithm 3. The latter, in turn, calls the subroutine `Apply_Intelim` described in Algorithm 4.

ALGORITHM 1 Decision procedure for 0-depth refutability

Input: A finite set X of S-formulae;

build the subformula graph G for X^u;
set $\mathsf{S} \in \lambda(A)$ for each $\mathsf{S}A \in X$;
set $\lambda(B) = \emptyset$ for each $B \notin X^u$;
`expanded _ graph` $=$ `Expand`$(\langle G, \lambda\rangle)$;
if `expanded _ graph` $= \langle G, \top\rangle$, **then**
  return true;
else
  return false;
end if

ALGORITHM 2 Decision procedure for 0-depth deducibility

Input: A finite set X of S-formulae and a S-formula φ;

build the subformula graph G for $X^u \cup \{\varphi^u\}$;
set $\mathsf{S} \in \lambda(A)$ for each $\mathsf{S}A \in X$;
set $\lambda(B) = \emptyset$ for each $B \notin X^u$;
`expanded _ graph` $=$ `Expand`$(\langle G, \lambda\rangle)$;
if `expanded _ graph` $= \langle G, \top\rangle$, **then**
  return true;
else if `expanded _ graph` $= \langle G, \lambda'\rangle$ and for S such that $\mathsf{S}\varphi^u := \varphi$, $\mathsf{S} \in \lambda'(\varphi^u)$, **then**
  return true;
else
  return false;
end if

ALGORITHM 3 `Expand`$(\langle G, \lambda\rangle)$

push all formulae A such that $\lambda(A) \neq \emptyset$ into `formula_stack`;
while $\lambda \neq \top$ and `formula_stack` is not empty **do**
  pop a formula A from `formula_stack`;
  `Apply_Intelim`$(A, \langle G, \lambda\rangle)$
end while

The correctness of both decision procedures follows from the fact that they return `true` iff the set $\{\mathsf{T}A \mid \mathsf{T} \in \lambda(A)\} \cup \{\mathsf{F}A \mid \mathsf{F} \in \lambda(A)\} \cup \{\mathsf{T}^*A \mid \mathsf{T}^* \in \lambda(A)\} \cup \{\mathsf{F}^*A \mid \mathsf{F}^* \in \lambda(A)\}$ is intelim saturated.

ALGORITHM 4 `Apply_Intelim`$(A, \langle G, \lambda \rangle)$

1: **for all** the G-modules M containing A **do**
2: let B be the top formula of M;
3: consider the set $Z = \{\mathsf{T}\, C \mid C \in M \text{ and } \mathsf{T} \in \lambda(C)\} \cup \{\mathsf{F}\, C \mid C \in M \text{ and } \mathsf{F} \in \lambda(C)\} \cup \{\mathsf{T}^*\, C \mid C \in M \text{ and } \mathsf{T}^* \in \lambda(C)\} \cup \{\mathsf{F}^*\, C \mid C \in M \text{ and } \mathsf{F}^* \in \lambda(C)\}$;
4: **if** Z contains all the premises for an application of an introduction rule with conclusion $\mathsf{S}B$ **then**
5: **if** $\bar{\mathsf{S}} \in \lambda(B)$ **then**;
6: **return** $\lambda = \top$
7: **else if** $\bar{\mathsf{S}} \notin \lambda(B)$ **then**
8: set $\mathsf{S} \in \lambda(B)$ and push B into `formula_stack`;
9: **else**
10: do nothing;
11: **end if**
12: **else**
13: **for all** $C \in M$ **do**
14: **if** Z contains all the premises for an application of an elimination rule with major premise $\mathsf{S}B$ and conclusion $\mathsf{S}C$, **then**
15: **if** $\bar{\mathsf{S}} \in \lambda(C)$ **then**;
16: **return** $\lambda = \top$
17: **else if** $\bar{\mathsf{S}} \notin \lambda(B)$ **then**
18: set $\mathsf{S} \in \lambda(C)$ and push C into `formula_stack`;
19: **else**
20: do nothing;
21: **end if**
22: **end if**
23: **end for**
24: **end if**
25: **end for**

We omit a detailed proof and just briefly discuss their complexity. The input for Algorithm 1 is a list of all the S-formulae in X, while the input for Algorithm 2 is a pair consisting of a list of all the S-formulae in X and the S-formula φ. Let n be the total size of the input; namely $O(|X|)$ for Algorithm 1 and $O(|X \cup \{\varphi\}|)$ for Algorithm 2. Observe that:

1. The number of nodes in the subformula graph G (line 1 of both decision procedures) is $O(n)$.
2. The cost of building the initial labeled subformula graph $\langle G, \lambda \rangle$ is $O(n^2)$.
3. The **while** loop in the `Expand` subroutine, in Algorithm 3, is executed at most as many times as there are nodes in G; namely $O(n)$ times. Here the key observation is that, in line 3, the formula A can be safely removed from `formula_stack`; i.e. each formula in `formula_stack` needs to be visited at most twice.
4. The essential cost of each run of the **while** loop consists in the cost of the `Apply_Intelim` subroutine (Algorithm 4).
5. For each formula A in G there are at most $O(n)$ G-modules containing A (line 1 of Algorithm 4).
6. The maximum number of nodes in a G-module is $a + 1$, where a is the maximum arity of a connective.

7. The cost of each run of the **forall** loop in the `Apply _ Intelim` subroutine is $O(a)$.

It follows from (1)–(7) that:

THEOREM 6.9
Whether or not $X \vdash_0 \varphi$ ($X \vdash_0$) can be decided in time $O(n^2)$, where $n = |X \cup \{\varphi\}|$ ($n = |X|$).

Now, a decision procedure for $X \vdash_k^{\mathsf{sub}} \varphi$ can be obtained from that of $X \vdash_0 \varphi$ by generalizing the `Expand` sobroutine (Algorithm 3) into the one displayed in Algorithm 5.

ALGORITHM 5 `Depth-Expand`$(k, \langle G, \lambda \rangle)$

if $k = 0$ **then**;
  `Expand`$(\langle G, \lambda \rangle)$;
else
  push all formulae A such that $\lambda(A) = \emptyset$ into `undefined_formulae`;
  while $\lambda \neq \top$ and `undefined_formulae` is not empty **do**
    pop a formula A from `undefined_formulae`;
    let λ_1 be such that $\mathsf{T} \in \lambda_1(A)$ (alternatively, $\mathsf{T}^* \in \lambda_1(A)$) and $\lambda_1(B) = \lambda(B)$ for all $B \neq A$;
    let λ_2 be such that $\mathsf{F} \in \lambda_2(A)$ (correspondingly, $\mathsf{F}^* \in \lambda_2(A)$) and $\lambda_2(B) = \lambda(B)$ for all $B \neq A$;
    let $G_1 =$ `Depth-Expand`$(k-1, \langle G, \lambda_1 \rangle)$;
    let $G_2 =$ `Depth-Expand`$(k-1, \langle G, \lambda_2 \rangle)$;
    if $\lambda_1 = \top$ **then**;
      set $\lambda = \lambda_2$;
      remove from `undefined_formulae` the formulae A such that $\lambda(A) \neq \emptyset$;
    else
      if $\lambda_2 = \top$ **then**;
        set $\lambda = \lambda_1$;
        remove from `undefined_formulae` the formulae A such that $\lambda(A) \neq \emptyset$;
      else
        set $\lambda(A) = x$ for all A such that $\lambda_1(A) = \lambda_2(A)$;
        remove from `undefined_formulae` the formulae A such that $\lambda(A) \neq \emptyset$;
      end if
    end if
  end while
end if

Then, a simple analysis shows that:

THEOREM 6.10
Whether or not $X \vdash_k^{\mathsf{sub}} \varphi$ ($X \vdash_k^{\mathsf{sub}}$) can be decided in time $O(n^{k+2})$, where $n = |X \cup \{\varphi\}|$ ($n = |X|$).

HINT.From Definitions 11 and the observation that there are $O(n)$ distinct subformulae of $X^u \cup \{\varphi^u\}$ (X^u). It can be shown that the procedure terminates in a number of steps less than or equal to

$$4^k \cdot \binom{n}{k} O(n^2) = O(n^{k+2}).$$

Note that the $4^k \cdot \binom{n}{k} O(n^2)$ is $O(c^n)$ when $k = n$, so that the upper bound for the unbounded system is just exponential. □

The above theorem refers to the basic case where the function that defines the virtual space is sub, but is not difficult to generalize it for any polynomially bounded virtual space [see 30, 31]. More precisely, when $f \trianglelefteq \mathsf{sub}$, the complexity of the decision problem is $O(n^{k+2})$. In general, the complexity is $O(p(n)^{k+2})$ where p is a polynomial depending on f (recall that, by definition, the virtual space is polynomially bounded).

7 5-valued non-deterministic semantics

The signs of our intelim method can be taken as *imprecise* truth-values that intuitively encode partial information about the standard truth-values in **4** [see 7, 9]; namely, two-element sets of the standard truth-values:

$$\mathrm{t} = \{\mathbf{t}, \mathbf{b}\}, \mathrm{f} = \{\mathbf{f}, \mathbf{n}\}, \mathrm{t}^* = \{\mathbf{t}, \mathbf{n}\}, \mathrm{f}^* = \{\mathbf{f}, \mathbf{b}\}.$$

Note that $\mathrm{t} \cap \mathrm{t}^* = \mathbf{t}$ and $\mathrm{f} \cap \mathrm{f}^* = \mathbf{f}$. Let us denote the set of these imprecise truth-values, $\{\mathrm{t}, \mathrm{f}, \mathrm{t}^*, \mathrm{f}^*\}$, by *4*. Now, we can take the elements of *4* as primitive, and use Dunn-style relational semantics [42] to define analogous notions to those of set-up and **4**-valuation:

DEFINITION 7.1
A *4-valuation* is a relation $\eta \subseteq At(\mathcal{L}) \times \mathit{4}$ such that:

i. for no p, $\langle p, \mathrm{t}\rangle$ and $\langle p, \mathrm{f}\rangle$ are both in η;
ii. for no p, $\langle p, \mathrm{t}^*\rangle$ and $\langle p, \mathrm{f}^*\rangle$ are both in η.

Given a *4*-valuation, η, this is extended to a relation $\eta \subseteq F(\mathcal{L}) \times \mathit{4}$ by recursive clauses:

$$\neg A\eta\mathrm{t} \text{ iff } A\eta\mathrm{f}^*;$$
$$\neg A\eta\mathrm{t}^* \text{ iff } A\eta\mathrm{f};$$
$$A \wedge B\eta\mathrm{t} \text{ iff } A\eta\mathrm{t} \text{ and } B\eta\mathrm{t};$$
$$A \wedge B\eta\mathrm{t}^* \text{ iff } A\eta\mathrm{t}^* \text{ and } B\eta\mathrm{t}^*;$$
$$A \wedge B\eta\mathrm{f} \text{ iff } A\eta\mathrm{f} \text{ or } B\eta\mathrm{f};$$
$$A \wedge B\eta\mathrm{f}^* \text{ iff } A\eta\mathrm{f}^* \text{ or } B\eta\mathrm{f}^*;$$
$$A \vee B\eta\mathrm{t} \text{ iff } A\eta\mathrm{t} \text{ or } B\eta\mathrm{t};$$
$$A \vee B\eta\mathrm{t}^* \text{ iff } A\eta\mathrm{t}^* \text{ or } B\eta\mathrm{t}^*;$$
$$A \vee B\eta\mathrm{f} \text{ iff } A\eta\mathrm{f} \text{ and } B\eta\mathrm{f};$$
$$A \vee B\eta\mathrm{f}^* \text{ iff } A\eta\mathrm{f}^* \text{ and } B\eta\mathrm{f}^*.$$

Intuitively, our imprecise values model a database which has only partial information of (or access to) a set of sources Ω. Let us say that an *agent's database* is a pair $D = \langle \Omega, \phi\rangle$, where Ω is a set of information sources and ϕ is a partial function $At(\mathcal{L}) \times \Omega \longrightarrow \{true, false\}$. So, given a *4*-valuation and denoting an agent with x:

- $p\eta$t iff x holds that there is $s \in \Omega$ such that $\phi(p,s) = true$ but does not hold that there is no $s \in \Omega$ such that $\phi(p,s) = false$;
- $p\eta$f iff x holds that there is no $s \in \Omega$ such that $\phi(p,s) = true$ but does not hold that there is no $s \in \Omega$ such that $\phi(p,s) = false$;
- $p\eta$t* iff x holds that there is no $s \in \Omega$ such that $\phi(p,s) = false$ but does not hold that there is no $s \in \Omega$ such that $\phi(p,s) = true$;
- $p\eta$f* iff a holds that there is $s \in \Omega$ such that $\phi(p,s) = false$ but does not hold that there is no $s \in \Omega$ such that $\phi(p,s) = true$.[12]

The notion of *4*-valuation would yield yet another alternative semantics for *full* **FDE**. However, we introduce it here only as a first step towards devising a semantics for the depth-bounded approximations defined proof-theoretically above.

7.1 The 0-depth logic

Within our conceptual framework, the value of formulae may be completely *undefined* when the agent's information of Ω is insufficient to even establish any of the imprecise values. As mentioned above, there is no reason to assume that an agent is 'told' about the values of atoms only. In most practical contexts, it may well be that the sources inform the agent that a certain disjunction is true without informing her about which of the two disjuncts is the true one, or, analogously, that a certain conjunction is false without informing her about which of the two conjuncts is the false one.

We shall denote $A\eta\bot$ whenever η is *undefined* for A. It is technically convenient to treat $\bot$ as a fifth value and take it as equivalent to the set $\{\mathbf{t}, \mathbf{b}, \mathbf{f}, \mathbf{n}\}$. Intuitively, this undefined value stands for full indeterminacy or ignorance about the standard defined value of a formula, which amounts to the situation where all those defined values are admissible. This in the sense the agent's information is not sufficient to discard even a single defined value. An important intuition here is that $\bot$ may eventually turn into an imprecise or even an standard truth-value by the development of the agent's reasoning or querying process. Thus, let us denote by *5* the set consisting of the elements of *4* together with $\bot$.

DEFINITION 7.2
A *5 non-deterministic valuation* is a relation $\eta' : F(\mathcal{L}) \times 2^{5}$ such that:
For no formula A, and $S_1, S_2 \in 2^{5}$, it is the case that:

i. $A\eta' S_1, A\eta' S_2$ and $\{\text{t}, \text{f}\} \subseteq S_1 \cup S_2$;
ii. $A\eta' S_1, A\eta' S_2$ and $\{\text{t}^*, \text{f}^*\} \subseteq S_1 \cup S_2$.

[12] Similar approaches to **FDE** are given in [7, 16, 17, 70], but they are extended along very different lines and used for very different purposes. Particularly, in those approaches there is no attempt to provide tractable approximations. We thank Luis Estrada-González for having pointed us at [70].

Moreover:

$$\neg A\eta'\{f^*\} \text{ iff } A\eta t;$$
$$\neg A\eta'\{t^*\} \text{ iff } A\eta f;$$
$$\neg A\eta'\{f\} \text{ iff } A\eta t^*;$$
$$\neg A\eta'\{t\} \text{ iff } A\eta f^*;$$
$$\neg A\eta'\{\perp\} \text{ iff } A\eta\perp;$$
$$A \wedge B\eta'\{f\} \text{ iff } A\eta f \text{ or } B\eta f;$$
$$A \wedge B\eta'\{f^*\} \text{ iff } A\eta f^* \text{ or } B\eta f^*;$$
$$A \wedge B\eta'\{\mathbf{f}\} \text{ iff } A\eta f \text{ and } B\eta f^*, \text{ or } A\eta f^* \text{ and } B\eta f;$$
$$A \wedge B\eta'\{t\} \text{ iff } A\eta t \text{ and } B\eta t;$$
$$A \wedge B\eta'\{t^*\} \text{ iff } A\eta t^* \text{ and } B\eta t^*;$$
$$A \wedge B\eta'\{\perp\} \text{ iff } A\eta t \text{ and } B\eta t^*, \text{ or } A\eta t^* \text{ and } B\eta t;$$
$$A \wedge B\eta'\{\perp, f^*\} \text{ iff } A\eta t \text{ and } B\eta\perp, \text{ or} A\eta\perp \text{ and } B\eta t;$$
$$A \wedge B\eta'\{\perp, f\} \text{ iff } A\eta t^* \text{ and } B\eta\perp, \text{ or } A\eta\perp \text{ and } B\eta t^*;$$
$$A \wedge B\eta'\{f, f^*, \perp\} \text{ iff } A\eta\perp \text{ and } B\eta\perp;$$
$$A \vee B\eta'\{t\} \text{ iff } A\eta t \text{ or } B\eta t;$$
$$A \vee B\eta'\{t^*\} \text{ iff } A\eta t^* \text{ or } B\eta t^*;$$
$$A \vee B\eta'\{\mathbf{t}\} \text{ iff } A\eta t \text{ and } B\eta t^*, \text{ or } A\eta t^* \text{ and } B\eta t;$$
$$A \vee B\eta'\{f\} \text{ iff } A\eta f \text{ and } B\eta f;$$
$$A \vee B\eta'\{f^*\} \text{ iff } A\eta f^* \text{ and } B\eta f^*;$$
$$A \vee B\eta'\{\perp\} \text{ iff } A\eta f \text{ and } B\eta f^*, \text{ or } A\eta f^* \text{ and } B\eta f;$$
$$A \vee B\eta'\{\perp, t^*\} \text{ iff } A\eta f \text{ and } B\eta\perp, \text{ or } A\eta\perp \text{ and } B\eta f;$$
$$A \vee B\eta'\{\perp, t\} \text{ iff } A\eta f^* \text{ and } B\eta\perp, \text{ or} A\eta\perp \text{ and } B\eta f^*;$$
$$A \vee B\eta'\{t, t^*, \perp\} \text{ iff } A\eta\perp \text{ and } B\eta\perp.$$

REMARK 7.3

The elements of the images of η' are computed via the standard **FDE**-tables as follows. Take the imprecise values as two-element sets of standard values and $\perp$ as equivalent to $\{\mathbf{t}, \mathbf{b}, \mathbf{f}, \mathbf{n}\}$. Apply the corresponding standard connective's function (Table 1) to the elements of the ordered pairs resulting from the Cartesian product of the sets corresponding to the values of the 'arguments' according to η (in the case of $\neg$, we apply the standard connective's function directly on the elements of the respective set). Then, considering only ordered pairs with the same first element, if the set collecting the values so obtained is equivalent to an imprecise value or $\perp$, take the latter as an element of the image of η'. However, if either sets equivalent to t *and* f, or to t* *and* f* are obtained, take respectively the union of those sets (i.e. $\perp$) as an element of the image of η'. Besides, if two singleton sets are

TABLE 4. 5N-tables

$\widetilde{\vee}$	t	f	t*	f*	$\perp$
t	{t}	{t}	{**t**}	{t}	{t}
f	{t}	{f}	{t*}	{$\perp$}	{$\perp$, t*}
t*	{**t**}	{t*}	{t*}	{t*}	{t*}
f*	{t}	{$\perp$}	{t*}	{f*}	{$\perp$, t}
$\perp$	{t}	{$\perp$, t*}	{t*}	{$\perp$, t}	{t, t*, $\perp$}

$\widetilde{\wedge}$	t	f	t*	f*	$\perp$
t	{t}	{f}	{$\perp$}	{f*}	{$\perp$, f*}
f	{f}	{f}	{f}	{**f**}	{f}
t*	{$\perp$}	{f}	{t*}	{f*}	{$\perp$, f}
f*	{f*}	{**f**}	{f*}	{f*}	{f*}
$\perp$	{$\perp$, f*}	{f}	{$\perp$, f}	{f*}	{f, f*, $\perp$}

$\widetilde{\neg}$	
t	f*
f	t*
t*	f
f*	t
$\perp$	$\perp$

obtained, form their union, which is equivalent to either an imprecise value or again to a singleton set. In the former case, take such an imprecise value as an element of the image of η'; in the latter case, take such a singleton set as equivalent to the image of η'.

These computations are justified by our intelim rules which, as explained above, are semantic in nature. For example, when $\vee$ is the connective at issue, and f and $\perp$ are respectively the values of the first and the second 'arguments' according to η, the elements of the image of η' can be only t* and $\perp$. For the other values are excluded on the basis of the rules: t cannot be in the image of η' since, otherwise, the value of the second 'argument' should have been t; f is also excluded because, otherwise, the values of both 'arguments' should have been f; similarly, f* is excluded since, otherwise, the values of both 'arguments' should have been f*. So, t* and $\perp$ are the only admissible values in the image of η'. In general, the elements of the images of η' can be obtained by excluding values in *5* on the basis of the rules, as the reader can verify.

Now, this 5-valued relational semantics *à la* Dunn can be summarized by the 5-valued non-deterministic truth-tables (5N-tables, for short) in Table 4. Thereby, we are in a position to introduce the following notion:

DEFINITION 7.4
Let $\mathcal{M}_{dfde}$ be the Nmatrix for $\mathcal{L}$, where $\mathcal{V} = 5$, $\mathcal{D} = \{t\}$ and the functions in $\mathcal{O}$ are defined by the 5N-tables in Table 4.[13]

[13] Owing to the logical symmetry between **b** and **n**, we can alternatively take $\mathcal{D} = \{t^*\}$.

Therefore, using the 5N-tables, a 5 non-deterministic valuation can be defined as a function. Namely, a *5-valuation* for $\mathcal{L}$ is a function $v : F(\mathcal{L}) \longrightarrow 5$. Then, we pick out from the set of all 5-valuations those which agree with the intended meaning of the connectives via $\mathcal{M}_{dfde}$:

DEFINITION 7.5
A *5N-valuation* is a 5-valuation v such that for all $A, B \in F(\mathcal{L})$:

1. $v(\neg A) = \widetilde{\neg}(v(A))$;
2. $v(A \circ B) \in \widetilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$ or $\wedge$.

REMARK 7.6
A 5N-valuation can be seen as describing an *information state* that is closed under the implicit information that depends only on the informational meaning of the connectives. This is information that the agent holds and with which he can operate, in the precise sense that he has a feasible procedure to decide, for every A, whether the information that A is t or f (analogously, t* or f*), or neither of them actually belongs to his information state.

DEFINITION 7.7
Given a 5N-valuation v, we say that a formula A is:

- *at least true* under v iff $v(A) = \mathrm{t}$;
- *non-true* under v iff $v(A) = \mathrm{f}$;
- *non-false* under v iff $v(A) = \mathrm{t}^*$;
- *at least false* under v iff $v(A) = \mathrm{f}^*$.

DEFINITION 7.8
A 5N-valuation v *realizes* a S-formula

- $\mathsf{T}A$ iff A is at least true under v;
- $\mathsf{F}A$ iff A is non-true under v;
- T^*A iff A is non-false under v;
- F^*A iff A is at least false under v.

A set X is said to be *5N-realizable* if there is a 5N-valuation v which realizes every element of X.

DEFINITIONS 13
For all X, φ,

- φ is a *0-depth consequence* of X, $X \vDash_0 \varphi$, iff for every 5N-valuation v, v realizes φ whenever v realizes all the elements of X;
- X is *0-depth inconsistent*, $X \vDash_0$, iff it is not 5N-realizable.

Analogously to the explosivity of the proof-theoretic characterization above, this explosivity regards inconsistent sets of *signed* formulae. Recall our explanation above of why this is compatible with the non-explosivity regarding (unsigned) formulae.

EXAMPLE 1
$\{\mathsf{T}\neg(A \vee B), \mathsf{T}\neg C\} \vDash_0 \mathsf{T}\neg(A \vee C)$ By inspection of the 5N-tables, it is easy to check that for any 5N-valuation v s.t. v realizes both $\mathsf{T}\neg(A \vee B)$ and $\mathsf{T}\neg C$, then v also realizes $\mathsf{T}\neg(A \vee C)$.

EXAMPLE 2
$\{\mathsf{T}\,(A \vee B) \wedge \neg A\} \nvDash_0 \mathsf{T}\,B \vee (A \wedge \neg A)$ Let v be a 5N-valuation s.t. $v(A) = \mathrm{f}^*$, $v(B) = \bot$, $v(A \vee B) = v(\neg A) = v((A \vee B) \wedge \neg A) = \mathrm{t}$, $v(A \wedge \neg A) = \mathrm{f}^*$ and $v(B \vee (A \wedge \neg A)) = \bot$.

EXAMPLE 3
$\{\mathsf{T}\,\neg(A \wedge B)\} \nvDash_0 \mathsf{T}\,\neg A \vee \neg B$ Let v be a 5N-valuation s.t. $v(A) = \mathrm{t}$, $v(B) = v(\neg B) = \bot$, $v(A \wedge B) = v(\neg A) = \mathrm{f}^*$, $v(\neg(A \wedge B)) = \mathrm{t}$ and $v(\neg A \vee \neg B) = \bot$.

EXAMPLE 4
$\{\mathsf{T}\,(A \vee B) \wedge \neg A, \mathsf{F}\,B \vee (A \wedge \neg A)\} \vDash_0$ By inspection of the 5N-tables, it is easy to check that there is no 5N-valuation v which realizes both $\mathsf{T}\,(A \vee B) \wedge \neg A$ and $\mathsf{F}\,B \vee (A \wedge \neg A)$.

EXAMPLE 5
$\{\mathsf{T}\,A \vee (B \wedge C), \mathsf{F}\,(A \vee B) \wedge (A \vee C)\} \nvDash_0$ Let v be a 5N-valuation s.t. $v(A) = \mathrm{f}^*$, $v(B) = v(A \vee B) = \mathrm{t}^*$, $v(C) = v(A \vee C) = v(B \wedge C) = \bot$, $v(A \vee (B \wedge C)) = \mathrm{t}$, and $v((A \vee B) \wedge (A \vee C)) = \mathrm{f}$.

Let us now show the adequacy of our informational 5-valued non-deterministic semantics with respect to the relation $\vdash_0$.

PROPOSITION 7.9
For all X and φ,

$$X \vDash_0 \varphi \text{ iff } X \vdash_0 \varphi.$$

PROOF. The soundness of the intelim rules can be immediately verified by inspection of the 5N-tables: every 5N-valuation which realizes the premise(s) of an intelim rule realizes also the conclusion of the rule. For example, if an agent holds the information that both A and B are t^*, then he also holds the information that $A \wedge B$ is t^*, since the 5N-table for $\wedge$ excludes the other imprecise values. Thereby, it follows by an elementary inductive argument that, if a 5N-valuation v realizes all the initial S-formulae of a 0-depth intelim tree $\mathcal{T}$ (i.e. an intelim sequence), then v realizes all the S-formulae occurring in $\mathcal{T}$. But, of course, no 5N-valuation can realize two conjugate S-formulae simultaneously. Thus, if $\mathcal{T}$ is a closed intelim tree, no 5N-valuation can realize all the initial S-formulae of $\mathcal{T}$. Therefore, on the one hand, if $\mathcal{T}$ is a 0-depth proof of φ from X, then for every 5N-valuation v, v realizes φ whenever v realizes all the elements of X. On the other hand, if $\mathcal{T}$ is a 0-depth refutation of X, then no 5N-valuation v realizes all the elements of X.

As for completeness, suppose that $X \nvdash_0 \varphi$. Then X is not 0-depth refutable; otherwise, by definition of 0-depth intelim proof, it should hold that $X \vdash_0 \varphi$, contrary to our hypothesis. Next, consider the set $Y = \{\psi | X \vdash_0 \psi\}$. Since X is not 0-depth refutable, for no A, $\mathsf{S}A$ and $\bar{\mathsf{S}}A$ are both in Y. Then, it is not difficult to verify that the function v defined as follows:

$$v(A) = \begin{cases} \mathrm{t} & \text{if } \mathsf{T}\,A \in Y \\ \mathrm{f} & \text{if } \mathsf{F}\,A \in Y \\ \mathrm{t}^* & \text{if } \mathsf{T}^*\,A \in Y \\ \mathrm{f}^* & \text{if } \mathsf{F}^*\,A \in Y \\ \bot & \text{otherwise} \end{cases}$$

is a 5N-valuation. Here we just outline a typical case. Suppose $v(A) = v(B) = \bot$. Then, $\mathsf{F}\, A \vee B \notin Y$. Otherwise, if $\mathsf{F}\, A \vee B \in Y$ then, by definition of Y and by the corresponding elimination rule for $\vee$, $\mathsf{F}\, A$ and $\mathsf{F}\, B$ should also be in Y. Hence, by definition of v, $v(A) = v(B) = \mathrm{f}$, against our assumption. Thus, by the 5N-table for $\vee$, $v(A \vee B) \neq \mathrm{f}$. Analogously, $\mathsf{F}^*\, A \vee B \notin Y$. Otherwise, if $\mathsf{F}^*\, A \vee B \in Y$ then, $\mathsf{F}^*\, A$ and $\mathsf{F}^*\, B$ should also be in Y. So, by definition of v, $v(A) = v(B) = \mathrm{f}^*$, against our assumption. Then, by the 5N-table for $\vee$, $v(A \vee B) \neq \mathrm{f}^*$. On the other hand, $\mathsf{T}\, A \vee B$ or $\mathsf{T}^*\, A \vee B$, may or may not belong to Y, and so $v(A \vee B) = \mathrm{t}$, $v(A \vee B) = \mathrm{t}^*$, or $v(A \vee B) = \bot$. Finally, observe that: (i) $\psi \in Y$ for all $\psi \in X$ and so, by definition of v, v realizes all $\psi \in X$; (ii) by the hypothesis that $X \nvdash_0 \varphi$, $\varphi \notin Y$ and so v does not realize φ. Therefore, $X \nvDash_0 \varphi$. □

COROLLARY 7.10
For all X,

$$X \vDash_0 \text{ iff } X \vdash_0 .$$

7.2 *k-depth logics*

Examples 2 and 3 above are valid inferences in **FDE** that are not so in the 0-depth approximation. Again, the latter is simply the logic of deductive reasoning restricted to the use of actual information. For those valid inferences that cannot be justified solely by the meaning of the connectives—i.e. by the 5N-tables—the incorporation of virtual information is required. This is information that is not even potentially contained in the current information state. Accordingly, the k-depth logics, $k > 0$, require the simulation of virtual extensions of the current information state. These extensions are formally defined via the notion of *5-refinement* below, where we take the values in *5* as partially ordered by two relations: (i) $\preceq_a$ such that $x \preceq_a y$ (read 'x is less defined than, or equal to, y') iff $x = \bot$ or $x = y$ for $x, y \in \{\mathrm{t}, \mathrm{f}, \bot\}$; (ii) $\preceq_b$ such that $x \preceq_b y$ iff $x = \bot$ or $x = y$ for $x, y \in \{\mathrm{t}^*, \mathrm{f}^*, \bot\}$.

DEFINITION 7.11
Let v, w be 5N-valuations. Then, w is a *5-refinement* of v, $v \sqsubseteq_5 w$, iff $v(A) \preceq_a w(A)$ or $v(A) \preceq_b w(A)$ for all A.

Now, the following definitions mimics Definitions 11:

DEFINITION 14
For all X, φ, and for all $f \in \mathcal{F}$,

- $X \vDash^f_0 \varphi$ iff $X \vDash_0 \varphi$;
- for $k > 0$, $X \vDash^f_k \varphi$ iff $X \cup \{\psi\} \vDash^f_{k-1} \varphi$ and $X \cup \{\bar{\psi}\} \vDash^f_{k-1} \varphi$ for some $\psi^u \in f(X^u \cup \{\varphi^u\})$.

When $X \vDash^f_k \varphi$ ($X \vDash^f_k$), we say that φ is a *k-depth consequence of* X (*X is k-depth inconsistent*) *over the f-bounded virtual space*.

As Definition 11, the above covers the case of k-depth inconsistency by assuming $X \vDash^f_k$ as equivalent to $X \vDash^f_k \varphi$ for all φ. Moreover, according with the above definitions, $X \vDash^f_k \varphi$ iff by simulating *either* a pair of refinements in which the truth-value of some A (in the virtual space defined by f) is respectively t or f, *or* a pair of refinements in in which the truth-value of some A is respectively t^* or f^*, the conclusion φ is realized by either of the members of the pair at depth

$k-1$.[14] That use of a defined truth-value for A, which is not even potentially contained in the current information state, is what we call virtual information.

EXAMPLE 6
$\{\mathsf{T}\,(A\vee B)\wedge\neg A\} \vDash_1^{\mathsf{sub}} \mathsf{T}\,B\vee(A\wedge\neg A)$ It is easy to check that $\{\mathsf{T}\,(A\vee B)\wedge\neg A\}\cup\{\mathsf{S}B\} \vDash_0 \mathsf{T}\,B\vee(A\wedge\neg A)$, and $\{\mathsf{T}\,(A\vee B)\wedge\neg A\}\cup\{\bar{\mathsf{S}}B\} \vDash_0 \mathsf{T}\,B\vee(A\wedge\neg A)$.

EXAMPLE 7
$\{\mathsf{T}\,\neg(A\wedge B)\} \vDash_1^{\mathsf{sub}} \mathsf{T}\,\neg A\vee\neg B$ It is easy to check that $\{\mathsf{T}\,\neg(A\wedge B)\}\cup\{\mathsf{T}^*\,A\} \vDash_0 \mathsf{T}\,\neg A\vee\neg B$ and $\{\mathsf{T}\,\neg(A\wedge B)\}\cup\{\mathsf{F}^*\,A\} \vDash_0 \mathsf{T}\,\neg A\vee\neg B$.

EXAMPLE 8
$\{\mathsf{T}\,A\vee(B\wedge C), \mathsf{F}\,(A\vee B)\wedge(A\vee C)\} \vDash_1^{\mathsf{sub}}$ It is easy to check that $\{\mathsf{T}\,A\vee(B\wedge C), \mathsf{F}\,(A\vee B)\wedge(A\vee C)\}\cup\{\mathsf{T}\,A\} \vDash_0$ and $\{\mathsf{T}\,A\vee(B\wedge C), \mathsf{F}\,(A\vee B)\wedge(A\vee C)\}\cup\{\mathsf{F}\,A\} \vDash_0$.

Now, the next proposition follows from the fact that RE_{fde} is sound and complete for full **FDE** [26]:

PROPOSITION 7.13
For all X, φ, and all $f \in \mathcal{F}$,

$$X \vDash_k^f \varphi \text{ iff } X \vdash_k^f \varphi.$$

The above given that RE_{fde} is a subsystem of our intelim method for unbounded k; i.e. a subsystem of the system constituted by the intelim rules together with an arbitrary number of applications of PB and PB^*. Indeed, the elimination rules together with PB and PB^* can be used to simulate any of the introduction rules. Conversely, the introduction rules together with PB and PB^* can be used to simulate any of the elimination rules. This clearly implies that the direct-proof system constituted by the introduction rules together with PB and PB^* —let us call it RI_{fde}—is also complete for full **FDE**. Nevertheless, there are two reasons for using both introduction and elimination rules: (i) it allows for more natural and shorter proofs, although not essentially shorter because the corresponding simulation is polynomial; (ii) it reduces the number of applications of PB and PB^* that, as stated above, is key to define the *depth* of an inference. In fact, regarding (i), it is not difficult to show that:[15]

PROPOSITION 7.14 ([72]).
RE_{fde} and RI_{fde} can linearly simulate each other. Moreover, the simulation preserves the subformula property.

[14] Analogously to Definitions 11, an S-formula φ may be realized *at certain depth* by one of those pairs of refinements but not by the other.

[15] The notions of RE_{fde}-tree (-refutation) and RI_{fde}-tree (-proof) should be clear. See [26, 72] for definitions.

TABLE 5. **LP**/$\mathbf{K_3}$-tables

$\tilde{\vee}$	*true*	*false*	*i*
true	*true*	*true*	*true*
false	*true*	*false*	*i*
i	*true*	*i*	*i*

$\tilde{\wedge}$	*true*	*false*	*i*
true	*true*	*false*	*i*
false	*false*	*false*	*false*
i	*i*	*false*	*i*

$\tilde{\neg}$	
true	*false*
false	*true*
i	*i*

$\tilde{\to}$	*true*	*false*	*i*
true	*true*	*false*	*i*
false	*true*	*true*	*true*
i	*true*	*i*	*i*

8 Depth-bounded approximations to LP and $\mathbf{K_3}$

8.1 Informational interpretation and need for imprecise values

It is well-known that **FDE**, **LP** and $\mathbf{K_3}$ are closely related to each other [see 2, 44, 45, 64]. As mentioned above, for a matrix to handle information that might be both inconsistent and partial, the availability of at least 4 different values is required. The matrix inducing **FDE** is an elegant example of such a matrix. Now, 3-valued matrices can be used to handle either inconsistency or partiality of information, one at a time. An example of a logic characterized by a 3-valued matrix handling inconsistency of information is the Logic of Paradox (**LP**). This matrix, $\mathcal{M}_3^b$, is the matrix for $\mathcal{L} = \{\vee, \wedge, \neg, \to\}$ where $\mathcal{V} = \{true, false, i\}$, $\mathcal{D} = \{true, i\}$ and the functions in $\mathcal{O}$ are defined by the truth-tables in Table 5. Here, the values *true* and *false* are the classical ones, and *i* stands for *both* (*true* and *false*). **LP** was introduced for rather philosophical purposes. Although Asenjo introduced the logic itself first [6], Priest coined **LP** has a tool for handling some logical paradoxes involving sentences that, according to Priest's view, are simultaneously true and false [63]. So, Priest interprets the 'inconsistent' third value in an aletheic sense. However, **LP** can be plausibly interpreted along the lines of the standard informational semantics of **FDE**. Namely, *true* (*false*) can be interpreted as 'there is a source assenting to *p* and *there is no* source dissenting to *p*' ('there is a source dissenting to *p* and *there is no* source assenting to *p*'); whereas *i* can be interpreted as 'there is a source assenting to *p* and there is a source dissenting to *p*'.

In turn, an example of a logic characterized by a 3-valued matrix handling partiality of information is Strong Kleene Logic ($\mathbf{K_3}$). This matrix, $\mathcal{M}_3^n$, is the matrix for $\mathcal{L} = \{\vee, \wedge, \neg, \to\}$ where $\mathcal{V} = \{true, false, i\}$, $\mathcal{D} = \{true\}$ and the functions in $\mathcal{O}$ are defined by the truth-tables in Table 5. Indeed, the only difference between $\mathcal{M}_3^b$ and $\mathcal{M}_3^n$ is their set of designated values. Accordingly, although in $\mathcal{M}_3^n$ *true* and *false* are again the classical ones, *i* is interpreted differently; viz., it stands for *neither* (*true* nor *false*). $\mathbf{K_3}$ was introduced for purposes in computer science, or rather they would have been if computer science had existed back then. In $\mathbf{K_3}$, the 'indeterminate' third value was originally introduced to account for inferences involving sentences for which its truth (or falsity) might not be decided by means of a function. More specifically, the third truth-value originally stands for 'undecidable by the algorithms whether true or false'. So, an informational interpretation of the truth-values is favoured. Correspondingly, the third truth-value behaves in a way compatible with any increase in information: If the value of some atomic formula *p* changed from indeterminate to either true or false, the value of any formula with *p* as a component must never change from true to false nor vice versa. Kleene referred to this as regularity; nowadays this is phrased in terms of monotony in an ordering. Thus, interpreting $\mathbf{K_3}$ along the lines of the standard informational semantics of **FDE** is not only plausible but natural: *true* (*false*) can be interpreted as 'there is a source assenting to p' ('there is a source dissenting to p'); whereas i can be interpreted as '*there is no* source assenting to p and *there is no* source dissenting to p'.[16]

Now, the valuations and consequence relations associated to the matrices of at issue are defined as for any many-valued logic. For instance:

DEFINITION 8.1
A $\mathcal{M}_3^n$*-valuation* is a function $v : F(\mathcal{L}) \longrightarrow \{true, false, i\}$ such that for all A, B:

1. $v(\neg A) = \widetilde{\neg}(v(A))$;
2. $v(A \circ B) = \widetilde{\circ}(v(A), v(B))$.

Where $\circ$ is $\vee$, $\wedge$ or $\rightarrow$.

DEFINITION 8.2
$\Gamma \vDash_{\mathcal{M}_3^n} A$ iff for every $\mathcal{M}_3^n$-valuation v, if $v(\Gamma) = true$, then $v(A) = true$.

The notion of $\mathcal{M}_3^b$-valuation and the corresponding relation $\vDash_{\mathcal{M}_3^b}$ are defined analogously.

Regarding their complexity, both logics at issue are co-NP complete, and so also idealized models of how an agent *can* reason. That **LP** is co-NP complete can be shown analogously to Proposition 3.3 [see also 3];[17] whereas, that $\mathbf{K_3}$ is co-NP complete follows from Cook's result that **CPL** is co-NP complete [24] together with the following:

PROPOSITION 8.3
A is a classical tautology iff $(p_1 \vee \neg p_1) \wedge ... \wedge (p_n \vee \neg p_n) \vDash_{\mathcal{M}_3^n} A$, where $p_1, ..., p_n$ are the atoms occurring in A.

PROOF. By definition, A is a classical tautology iff for every classical valuation, $v : F(\mathcal{L}) \longrightarrow \{true, false\}$, $v(A) = true$. In turn, also by definition, this holds iff for every $\mathcal{M}_3^n$–valuation, $v : F(\mathcal{L}) \longrightarrow \{true, false, i\}$, if $v(p_i) \neq i$ for all p_i occurring in A, then $v(A) = true$. By the $\mathbf{K_3}$-tables for $\neg$ and $\vee$, this holds iff for every $\mathcal{M}_3^n$-valuation v, $v(A) = true$ whenever $v(p_i \vee \neg p_i) = true$ for all p_i occurring in A. In turn, by the $\mathbf{K_3}$-table for $\wedge$, this holds iff for every $\mathcal{M}_3^n$-valuation v, $v(A) = true$ whenever $v((p_1 \vee \neg p_1) \wedge ... \wedge (p_n \vee \neg p_n)) = true$ for all p_i occurring in A. Hence, the latter holds iff $(p_1 \vee \neg p_1) \wedge ... \wedge (p_n \vee \neg p_n) \vDash_{\mathcal{M}_3^n} A$. □

This again brings us to the need for tractable approximations. The basis for defining our approximations are sort of natural deduction systems based on observations analogous to those regarding **FDE**. Observe that, under the informational interpretation of **LP**, only the value i can be taken as *stable* without assuming complete information about the set of sources Ω. That is, given an epistemic state that evolves over time, the values *true* and *false* can be regarded as stable only if complete knowledge of Ω is assumed. Thus, these latter values are *information-transcendent* when interpreted as timeless, for they refer to an 'objective' informational situation concerning the domain of all sources. In a dual manner, under the informational interpretation of $\mathbf{K_3}$, the values *true* and *false* are stable without assuming full knowledge of Ω (since the possibility

[16] In $\mathbf{K_3}$ the possibility of contradictory information is discarded: once a source assents (dissents) to p, the possibility of there being a source dissenting (assenting) to p is discarded.

[17] In fact, a formula A is a tautology in **CPL** iff A is a tautology in **LP** [63].

of contradictory information is discarded); whereas the value i is information-transcendent when interpreted as timeless. Much as in the case of **FDE**, this situation motivates the need for *stable imprecise values*. Again, we address this question by shifting to signed formulae, where the signs express such imprecise values associated with two distinct bipartitions of the corresponding set of standard values.

8.2 *Intelim deduction in **LP** and* $\mathbf{K_3}$

We use the same signed formulae used for **FDE** but, of course, interpret them differently. To state such a re-interpretation, we use x to refer to an agent and v to denote respectively a $\mathcal{M}_3^b$-valuation or a $\mathcal{M}_3^n$-valuation. For **LP**, we interpret: $\mathsf{T}A$ as "x holds that A is at least true" (expressing that $v(A) \in \{true, i\}$); $\mathsf{F}A$ as 'x holds that A is false only' ($v(A) \in \{false\}$); T^*A as 'x holds that A is true only' ($v(A) \in \{true\}$); F^*A as 'x holds that A is at least false' ($v(A) \in \{false, i\}$). As for $\mathbf{K_3}$, we interpret: $\mathsf{T}A$ as "x holds that A is true" ($v(A) \in \{true\}$); $\mathsf{F}A$ as "x holds that A is non-true" ($v(A) \in \{false, i\}$); T^*A as 'x holds that A is non-false' ($v(A) \in \{true, i\}$); F^*A as 'x holds that A is false' ($v(A) \in \{false\}$). Crucially, according to the respective informational interpretation of the values of **LP** and $\mathbf{K_3}$, whereas S-formulae of the form $\mathsf{T}A$ and F^*A involve only information that does not require complete knowledge of the sources, S-formulae of the form T^*A and $\mathsf{F}A$ involve information that does require such a complete knowledge.

Thereby, by making minor modifications to our proof system for **FDE**, we can obtain systems for **LP** and $\mathbf{K_3}$ which naturally lead to defining analogous hierarchies of tractable depth-bounded approximations to the latter logics. Namely, those systems are obtained by enriching the rules of the **FDE**'s intelim method with the intelim rules for implication displayed in Table 6, together with, respectively, the following structural rules which clearly do not involve introducing virtual information:

$$\frac{\mathsf{T}^*A}{\mathsf{T}A} \qquad \frac{\mathsf{F}A}{\mathsf{F}^*A} \qquad\qquad\qquad \frac{\mathsf{T}A}{\mathsf{T}^*A} \qquad \frac{\mathsf{F}^*A}{\mathsf{F}A}$$

Additional rules for **LP** Additional rules for $\mathbf{K_3}$

In turn, hierarchies of depth-bounded approximations can be defined in terms of the maximum number of nested applications of *PB* and *PB**, exactly as before. Although *PB* and *PB** are not the unique structural rules in the case of **LP** and $\mathbf{K_3}$, they are the only rules involving the introduction of virtual information. Moreover, it is straightforward to adapt the proofs of Theorems 6.9 and 6.10 to show the tractability of the approximations at issue. Further, by making minor modifications to our semantical framework for the hierarchy of approximations to **FDE**, it can be shown that the hierarchies for the 3-valued logics also admit of a 5-valued non-deterministic semantics.

9 Related and future work

In this section, we briefly recall some proof systems and a cluster of ideas closely related to our approach in this paper. However, in [72] it is shown that a crucial difference between the tableau methods to be recalled below and the intelim methods introduced above is that the latter have an exponential speed-up on the former. Namely, there we introduce a new class of examples which we prove to be hard for all tableau systems sharing the $\vee/\wedge$ rules with the classical one [see 7]—which include tableaux for the three many-valued logics treated in this paper—but easy for

TABLE 6. Intelim rules for the implication of **LP** and $\mathbf{K_3}$

$$\frac{\mathsf{F}^* A}{\mathsf{T} A \to B} \qquad \frac{\mathsf{T} B}{\mathsf{T} A \to B} \qquad \frac{\begin{array}{l}\mathsf{T}^* A\\ \mathsf{F} B\end{array}}{\mathsf{F} A \to B}$$

$$\frac{\mathsf{F} A}{\mathsf{T}^* A \to B} \qquad \frac{\mathsf{T}^* B}{\mathsf{T}^* A \to B} \qquad \frac{\begin{array}{l}\mathsf{T} A\\ \mathsf{F}^* B\end{array}}{\mathsf{F}^* A \to B}$$

$$\frac{\mathsf{F} A \to B}{\mathsf{T}^* A} \qquad \frac{\mathsf{F} A \to B}{\mathsf{F} B} \qquad \frac{\mathsf{F}^* A \to B}{\mathsf{T} A} \qquad \frac{\mathsf{F}^* A \to B}{\mathsf{F}^* B}$$

$$\frac{\begin{array}{l}\mathsf{T} A \to B\\ \mathsf{T}^* A\end{array}}{\mathsf{T} B} \qquad \frac{\begin{array}{l}\mathsf{T}^* A \to B\\ \mathsf{T} A\end{array}}{\mathsf{T}^* B} \qquad \frac{\begin{array}{l}\mathsf{T} A \to B\\ \mathsf{F} B\end{array}}{\mathsf{F}^* A} \qquad \frac{\begin{array}{l}\mathsf{T}^* A \to B\\ \mathsf{F}^* B\end{array}}{\mathsf{F} A}$$

their analogous intelim systems. Roughly, the reason of this is that while the tableau methods have operational branching rules that imply a good deal of redundant branchings in the corresponding tree, the intelim methods have only structural branching rules that reduce the amount of branching to a minimum by making all branches mutually exclusive. So, the intelim systems introduced in this paper are interesting independently of the depth-bounded approach. In turn, besides computational efficiency issues and within the context of the approach, another important difference between the tableau methods to be recalled below and the intelim methods introduced above is that since in the former cuts are eliminable, no approximations can be defined by controlling the application of the cut rules. Further, the natural deduction systems recalled below do not comply with a key idea underlying our depth-bounded approximations, according to which the meaning of a logical operator is fixed only in terms of actual information. For in the natural deduction systems, some of the (operational) 'discharge' rules make essential use of virtual information.

9.1 Tree methods, signs and natural deduction

The first tableaux for **FDE** are due to Dunn [42]. He introduced a direct-proof tableau system based on a modification of Jeffrey's method of (classical) 'coupled trees'. Dunn's system rules are syntactically identical to the rules of the *unsigned* version of Smullyan's classical tableaux [71]. Now, to the best of our knowledge, the first systems for **FDE** based on S-formulae—in which the signs stand for sets of values instead of single values—are two refutation tree systems due to D'Agostino [26]. Soon after, Fitting [45] introduced a direct-proof tableau system for **FDE** which is based on the same use of S-formulae. In fact, in the same period, a general method to use signs as sets of truth-values suitable for any finite-valued propositional logic was provided by Hähnle [48]. The key idea underlying that use is to increase the expressivity of the signs and, thus, significantly decrease the number of new branches per rule application. Later on, Avron [7] used four signs—interpreting them as intuitively corresponding to positive/negative information concerning truth/falsity—to provide tableaux for a diversity of logics; including **FDE**, all 3-valued logics, and some logics that do not have finite characteristic matrix.

The S-formulae underlying D'Agostino's systems are the same that we used for the systems introduced in this paper, with the caveat that in D'Agostino's those S-formulae are interpreted in terms of information that an agent possesses without any requirement of such an information being *actual*—in accordance with the interpretation of the standard 4 values. That is, $\mathsf{T}A$ is interpreted as '$v(A) = \mathbf{t}$ or $v(A) = \mathbf{b}$', $\mathsf{F}A$ as '$v(A) = \mathbf{f}$ or $v(A) = \mathbf{n}$', and so on. Thereby, the rules of the first system introduced by D'Agostino are formally analogous to the rules of the signed version of Smullyan's classical tableaux. Now, essentially the same tableau refutation system was reintroduced by Bloesch [18], who used it for both **FDE** and **LP**. Besides, a system differing only notationally and used for those paraconsistent logics as well as for $\mathbf{K_3}$, was given by Priest [64]. Further, the same system was used for **FDE** by Fitting in [46]. In turn, Fitting's first system for **FDE** in [45], although formulated using essentially the same S-formulae of the previous sections, is closer to Dunn's coupled trees method. Besides, in [45] only two signs, T and F, are used (explicitly) and a different convention on them is followed. Namely, a *signed formula* is an expression of the form $\mathsf{T}A$ or $\mathsf{F}A$, where A is a formula and which is intuitively and respectively interpreted as 'A is at most true'—meaning that $v(A) = \mathbf{t}$ or $v(A) = \mathbf{n}$—and 'A is at most false'—meaning that $v(A) = \mathbf{f}$ or $v(A) = \mathbf{n}$. Since only the signs T and F are used, the corresponding rules are syntactically identical to the rules of the signed version of Smullyan's classical tableaux.

As for D'Agostino's second refutation system for **FDE** in [26], it was introduced as a more computationally efficient alternative to other proof systems. It was baptized RE_{fde} since it is based on *KE* [see 26, 37]. The hallmark of *KE*—inherited by RE_{fde}—is the reduction of the amount of branching to a minimum by making all branches mutually exclusive. Accordingly, RE_{fde} has only two branching rules expressing a generalized rule of Bivalence (our *PB* and *PB** above) and the rest of its rules have all a linear format (our elimination rules in Table 3). In fact, as Propositions 7.13 and 7.14, as well as comments thereof suggest, RE_{fde} (equivalently, RI_{fde}) may well serve as the basis for defining depth-bounded approximations to **FDE**. The reason for using both introduction and elimination rules were stated above; however, as a refutation method, RE_{fde} may be still preferred for applications in automated reasoning. Further, of course, *KE*-like systems serving as basis for approximations to **LP** and $\mathbf{K_3}$ can be straightforwardly obtained by discarding the corresponding introduction rules.

Now, in the literature there are also natural deduction systems that are closely related to the systems that we introduced in this paper. However, all those natural deduction systems are formulated in terms of unsigned formulae. A Gentzen-Prawitz style system for **FDE**, **LP** and $\mathbf{K_3}$ was given by Priest in [65]. Alternative natural deduction systems for **FDE** were given by Voishvillo [76], and Tamminga and Tanaka [74]. Regarding **LP**, essentially the same system was reintroduced by Kooi and Tamminga [51]. As for $\mathbf{K_3}$, essentially the same system was introduced again by Tamminga [73]. Besides, alternative Fitch-style systems for the three logics were given by Roy [68].

9.2 *Depth-bounded reasoning and efficient proof systems*

Approximations to (fragments, full, or extensions of) **CPL** via tractable subsystems of increasing inferential power have been investigated since 1990s [e.g. 19, 25, 39, 53, 69]. A hierarchy of tractable depth-bounded approximations to **CPL**, based on an intelim method analogous to the systems presented in this paper, was widely studied in [e.g. 30, 31]. Such a hierarchy, as well as ideas thereof, have been further explored and applied in diverse works. For example: in [26, 37], proof-theoretic and computational advantages of the '*KE* fragment' of the corresponding classical intelim method over tableaux and cut-free sequent calculus are studied; in [28], it is outlined how depth-bounded classical reasoning provides means to solve the Bar-Hillel-Carnap paradox, the enduring scandal of deduction, and the problem of logical omniscience; in [36], some applications of the depth-bounded approach to classical logic in formal argumentation theory are studied; semantic bases to defining similar hierarchies of approximations to epistemic logics are put forward in [54]; a preliminary step for an extension to classical first-order logic is given in [35]; a multiagent setting has been explored in [23, 55]; an application for defining depth-bounded belief functions for reasoning with uncertainty was given in [13]; an application in the context of answer set programming was introduced in [12]; and a hierarchy based solely on *KE* has been put forward in [37, 72].

The novelty of the contribution of this paper—together with its preliminary version [38]—is that it starts exploring the extension of the depth-bounded approach to propositional *non-classical* logics. Namely, we defined hierarchies of tractable depth-bounded approximations to three many-valued logics which are closely related to each other. These hierarchies are analogous to the hierarchy of approximations to **CPL** and may be similarly further explored and applied. To begin with, as mentioned above, it is shown in [72] that the proof-systems underlying the hierarchies are more efficient than their tableau systems counterparts. Besides, extending the methodology used in this paper to a variety of finite-valued logics in the spirit of [20, 21, 49, 62] will be the topic of a subsequent paper. In fact [20] already paves the way for carrying out such an extension since it introduces a general method for extracting *KE*-style systems for any finite-valued logic. Those systems, as expected, are interesting in their own right because they outperform their tableau systems counterparts. Moreover, there is ongoing work on defining—both proof-theoretically and semantically—a hierarchy of depth-bounded approximations to intuitionistic propositional logic [72]. The latter will presumably pave the way for defining similar hierarchies for a wide variety of logics characterized by Kripke-style semantics.

Further, the approximations to the simple paraconsistent logics **FDE** and **LP** defined above might serve—both conceptually and technically—as a starting point to extend the approach to more refined paraconsistent logics such as the logics of formal inconsistency (**LFIs**) [22] and the logics of evidence and truth (**LETs**) [67]. For these extensions, the *KE*-style systems for **mbC**, **mCi** and $\boldsymbol{C_1}$ given in [59–61] could be useful. What is more, hierarchies of depth-bounded approximations to modal and substructural logics could be worked out via *KE*-style systems already provided for them in the literature [5, 33, 34, 43, 47]. In fact, depth-bounded approximations to normal modal logics would simultaneously constitute approximations to **LFIs**, given that the former can be rewritten as the latter [56]. In parallel to these extensions, major results would be showing that the underlying *KE*-style systems are indeed more efficient that their tableu systems counterparts [see 20], as such results are so far only conjectured. These major results seem attainable since, for instance, the examples given in [72], which are hard for all tableau systems sharing the $\vee/\wedge$ rules with the classical one but easy for their analogous intelim systems, should also work for the 'additive fragment' of substructural logics.

Thus, we envisage extensions of the depth-bounded approach covering an ample variety of non-classical logics. As well-known, these logics find natural applications in Computer Science, AI, Philosophy and Cognitive Science. Depth-bounded approximations to non-classical logics could then be used to provide models of tractable reasoning and computation in, for instance, medical and juridical expert systems [see 4, 61] as well as databases and knowledge-bases management [see 40]. In particular, the approximations introduced in this paper might be used to provide tractable models of inconsistency search [see 52] and of the deductive behavior of evidence [see 67]. Moreover, the systems underlying our hierarchies may be implemented in automated or interactive theorem provers—presumably, being more suitable the *KE*-style systems for the former [see 4, 58] and intelim-style systems for the latter [see 27]. Further, either kind of systems can yield proof search heuristics based on the choice of the formulae on which the corresponding branching rules are applied [see 37, 72].

Acknowledgements

We thank two anonymous reviewers for helpful comments and suggestions. Solares-Rojas is very much grateful to Carlos César Jiménez for encouraging his interest in the relation between logic and computation, and in pursuing studies abroad. Carlos left us too early, but his philosophical legacy is as alive as ever. This research was funded by the Department of Philosophy ‘Piero Martinetti’ of the University of Milan under the Project ‘Departments of Excellence 2023-2027’ and by the PRIN 2017 Project (cod. 20173YP4N3) awarded by the Ministry of University and Research (MUR). Solares-Rojas was partly supported by the PAPIIT project IG400422 ‘Propositional variables and propositional constants’.

References

[1] A. Anderson and N. Belnap. Tautological entailments. *Philosophical Studies*, **13**, 9–24, 1962.
[2] O. Arieli, A. Avron and A. Zamansky. *Theory of Effective Propositional Paraconsistent Logics, Volume 75 of Studies in Logic*. College Publications, London, 2018.
[3] O. Arieli and M. Denecker. Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, **13**, 557–580, 2003.
[4] A. Artosi, P. Cattabriga and G. Governatori. An automated approach to normative reasoning. In *ECAI 1994, Artificial Normative Reasoning*, pp. 132–145, Amsterdam, 1994.
[5] A. Artosi and G. Governatori. Labelled model modal logic. In *Proceedings of the CADE-12 Workshop on Automated Model Building*, pp. 11–17, 1994.
[6] F. Asenjo. A calculus of antinomies. *Notre Dame Journal of Formal Logic*, **7**, 103–105, 1966.
[7] A. Avron. Tableaux with four signs as a unified framework. In *Automated Reasoning with Analytic Tableaux and Related Methods, Int. Conf., TABLEAUX 2003, Volume 2796 of Lecture Notes in Computer Science*, M. Cialdea Mayer and F. Pirri, eds, pp. 4–16. Springer, Berlin, Heidelberg, 2003.
[8] A. Avron. A non-deterministic view on non-classical negations. *Studia Logica*, **80**, 159–194, 2005.
[9] A. Avron. Multi-valued semantics: Why and how. *Studia Logica*, **92**, 163–182, 2009.
[10] A. Avron and I. Lev. Non-deterministic multiple-valued structures. *Journal of Logic and Computation*, **15**, 241–261, 2005.
[11] A. Avron and A. Zamansky. Non-deterministic semantics for logical systems. In *Handbook of Philosophical Logic*, D. Gabbay and F. Guenthner, eds, vol. 16, pp. 227–304. Springer, Dordrecht, 2011.
[12] P. Baldi, F. D’Asaro and G. Primiero. Introducing k-lingo: A k-depth bounded version of ASP system clingo. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, vol. 18, pp. 661–665, 2021.
[13] P. Baldi and H. Hosni. Depth-bounded belief functions. *International Journal of Approximate Reasoning*, **123**, 26–40, 2020.
[14] N. Belnap. How a computer should think. In *Contemporary Aspects of Philosophy*, G. Ryle, ed, pp. 30–55. Oriel Press, Stocksfield, 1977.
[15] N. Belnap. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logics*, J. Dunn and G. Epstein, eds, pp. 5–37. Reidel Publishing Company, Dordrecht, 1977.

[16] C. Blasio. Up the hill: On the notion of information in logics based on the four-valued bilattice. In *2nd Workshop on Philosophy, Logic and Analytical Metaphysics (FILOMENA 2)*, p. 99, 2015.
[17] C. Blasio. Revisitando a lógica de Dunn-Belnap. *Manuscrito*, **40**, 99–126, 2017.
[18] A. Bloesch. A tableau style proof system for two paraconsistent logics. *Notre Dame Journal of Formal Logic*, **34**, 295–301, 1993.
[19] M. Cadoli and M. Schaerf. Tractable reasoning via approximation. *Artificial Intelligence*, **74**, 249–310, 1995.
[20] C. Caleiro, J. Marcos and M. Volpe. Bivalent semantics, generalized compositionality and analytic classic-like tableaux for finite-valued logics. *Theoretical Computer Science*, **603**, 84–110, 2015.
[21] W. Carnielli. Systematization of finite many-valued logics through the method of tableaux. *The Journal of Symbolic Logic*, **52**, 473–493, 1987.
[22] W. Carnielli, M. Coniglio and J. Marcos. Logics of formal inconsistency. In *Handbook of Philosophical Logic*, D. Gabbay and F. Guenthner, eds, pp. 1–93. Springer, Netherlands, Dordrecht, 2007.
[23] G. Cignarale and G. Primiero. A multi-agent depth bounded boolean logic. In *Software Engineering and Formal Methods. SEFM 2020, Volume 12524 of Lecture Notes in Computer Science*, L. Cleophas and M. Massink., eds, pp. 176–191. Springer, Cham, 2020.
[24] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pp. 151–158. Association for Computing Machinery, New York, 1971.
[25] J. Crawford and D. Etherington. A non-deterministic semantics for tractable inference. In *AAAI/IAAI*, pp. 286–291. AAAI Press / The MIT Press, 1998.
[26] M. D'Agostino. *Investigations into the Complexity of some Propositional Calculi*. Oxford University Computing Laboratory. Programming Research Group, 1990.
[27] M. D'Agostino. Classical natural deduction. In *We Will Show them!* S. Artëmov, H. Barringer, A. d'Avil, L. Lamb and J. Woods, eds, vol. 1, pp. 429–468. College Publications, London, 2005.
[28] M. D'Agostino. Depth-bounded logic for realistic agents. *L&PS–Logic & Philosophy of Science*, **11**, 3–57, 2013.
[29] M. D'Agostino. Analytic inference and the informational meaning of the logical operators. *Logique et Analyse*, **57**, 407–437, 2014.
[30] M. D'Agostino. An informational view of classical logic. *Theoretical Computer Science*, **606**, 79–97, 2015.
[31] M. D'Agostino, M. Finger and D. Gabbay. Semantics and proof-theory of depth bounded boolean logics. *Theoretical Computer Science*, **480**, 43–68, 2013.
[32] M. D'Agostino and L. Floridi. The enduring scandal of deduction. *Synthese*, **167**, 271–315, 2009.
[33] M. D'Agostino, D. Gabbay and K. Broda. Tableau methods for substructural logics. In *Handbook of Tableau Methods*, M. D'Agostino, D. Gabbay, R. Hähnle and J. Posegga, eds, pp. 397–467. Springer, Dordrecht, 1999.
[34] M. D'Agostino, D. Gabbay and A. Russo. Grafting modalities onto substructural implication systems. *Studia Logica*, **59**, 65–102, 1997.
[35] M. D'Agostino, C. Larese and S. Modgil. Towards depth-bounded natural deduction for classical first-order logic. *Journal of Applied Logics*, **2631**, 423–451, 2021.
[36] M. D'Agostino and S. Modgil. Classical logic, argument and dialectic. *Artificial Intelligence*, **262**, 15–51, 2018.

[37] M. D'Agostino and M. Mondadori. The taming of the cut. Classical refutations with analytic cut. *Journal of Logic and Computation*, **4**, 285–319, 1994.
[38] M. D'Agostino and A. Solares-Rojas. Towards tractable approximations to many-valued logics: The case of first degree entailment. In *The Logica Yearbook 2021*, I. Sedlár, ed, pp. 57–76. College Publications, 2022.
[39] M. Dalal. Anytime families of tractable propositional reasoners. *Annals of Mathematics and Artificial Intelligence*, **22**, 297–318, 1998.
[40] S. de Amo, W. Carnielli and J. Marcos. A logical framework for integrating inconsistent information in multiple databases. In *Foundations of Information and Knowledge Systems*, T. Eiter and K.-D. Schewe, eds, pp. 67–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
[41] M. Dummett. *The Logical Basis of Metaphysics*. Duckworth, London, 1991.
[42] J. Dunn. Intuitive semantics for first-degree entailments and 'coupled trees'. *Philosophical Studies*, **29**, 149–168, 1976.
[43] M. Finger. Analytic methods for the logic of proofs. *Journal of Logic and Computation*, **20**, 167–188, 2008.
[44] M. Fitting. Bilattices and the semantics of logic programming. *The Journal of Logic Programming*, **11**, 91–116, 1991.
[45] M. Fitting. Kleene's three valued logics and their children. *Fundamenta Informaticae*, **20**, 113–131, 1994.
[46] M. Fitting. *Primary Course Notes: First Degree Entailment, Continued*. The Graduate Center, City University of New York , April, 2018.
[47] G. Governatori and A. Luppi. Labelled tableaux for non-normal modal logics. In *AI*IA 99: Advances in Artificial Intelligence*, E. Lamma and P. Mello, eds, pp. 119–130. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
[48] R. Hähnle. Towards an efficient tableau proof procedure for multiple-valued logics. In *Computer Science Logic*, E. Börger, H. K. Büning, M. Richter and W. Schönfeld, eds, pp. 248–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991.
[49] R. Hähnle. Tableaux for many-valued logics. In *Handbook of Tableau Methods*, M. D'Agostino, D. Gabbay, R. Hähnle and J. Posegga, eds, pp. 529–580. Springer, Dordrecht, 1999.
[50] S. Kleene. *Introduction to Metamathematics*. Van Nostrand, Princeton, 1952.
[51] B. Kooi and A. Tamminga. Completeness via correspondence for extensions of the logic of paradox. *The Review of Symbolic Logic*, **5**, 720–730, 2012.
[52] D. Kozhemiachenko and L. Vashentseva. Non-contingency in a paraconsistent setting. *Logic Journal of the IGPL*, **01**, 2023.
[53] G. Lakemeyer and H. Levesque. A first-order logic of limited belief based on possible worlds. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, vol. 17, pp. 624–635, 2020.
[54] C. Larese. *The Principle of Analyticity of Logic.* PhD Thesis, Scuola Normale Superiore, Pisa, 2019.
[55] M. Larotonda and G. Primiero. A depth-bounded semantics for becoming informed. In *Software Engineering and Formal Methods. SEFM 2022 Collocated Workshops*, P. Masci, C. Bernardeschi, P. Graziani, M. Koddenbrock and M. Palmieri, eds, pp. 366–382. Springer International Publishing, Cham, 2023.
[56] J. Marcos. Nearly every normal modal logic is paranormal. *Logique et Analyse*, **48**, 279–300, 2005.
[57] M. Mondadori. Efficient inverse tableaux. *Logic Journal of the IGPL*, **3**, 939–953, 1995.
[58] A. Neto. *A Multi-Strategy Tableau Prover* . PhD Thesis, University of São Paulo, 2007.

[59] A. Neto and M. Finger. Effective prover for minimal inconsistency logic. In *Artificial Intelligence in Theory and Practice*, M. Bramer, ed., pp. 465–474. Springer US, Boston, MA, 2006.
[60] A. Neto and M. Finger. A KE tableau for a logic of formal inconsistency. In *Proceedings of TABLEAUX'07 Position Papers and Workshop on Agents, Logic and Theorem Proving. Technical Report (LSIS.RR.2007.002) of the LSIS/Université Paul Cézanne*, Marseille, France, 2007.
[61] A. Neto, C. Kaestner and M. Finger. Towards an efficient prover for the C_1 paraconsistent logic. *Electronic Notes in Theoretical Computer Science*, **256**, 87–102, 2009.
[62] F. Paoli. Tautological entailments and their rivals. In *Handbook of Paraconsistency*, J. Béziau, W. Carnielli and F. Gabbay, eds, pp. 153–175. College Publications, London, 2007.
[63] G. Priest. The logic of paradox. *Journal of Philosophical Logic*, **8**, 219–241, 1979.
[64] G. Priest. *An Introduction to Non-classical Logic*. Cambridge University Press, 2001.
[65] G. Priest. Paraconsistent logic. In *Handbook of Philosophical Logic*, D. Gabbay and F. Guenthner, eds, vol. 6, pp. 287–393. Springer, Dordrecht, 2002.
[66] W. Quine. *The Roots of Reference*. Open Court, LaSalle, 1973.
[67] A. Rodrigues, J. Bueno-Soler and W. Carnielli. Measuring evidence: A probabilistic approach to an extension of Belnap–Dunn logic. *Synthese*, **198**, 5451–5480, 2021.
[68] T. Roy. Natural derivations for Priest, an introduction to non-classical logic. *The Australasian Journal of Logic*, **4**, 47–192, 2006.
[69] M. Sheeran and G. Stålmarck. A tutorial on Stålmarck's proof procedure for propositional logic. *Formal Methods in System Design*, **16**, 23–58, 2000.
[70] Y. Shramko and H. Wansing. Some useful 16-valued logics: How a computer network should think. *Journal of Philosophical Logic*, **34**, 121–153, 2005.
[71] R. Smullyan. *First-Order Logic*. Springer, Berlin, 1968.
[72] A. Solares-Rojas. *Tractable Depth-Bounded Approximations to some Propositional Logics*. PhD Thesis, Univeristy of Milan, Milan, 2022.
[73] A. Tamminga. Correspondence analysis for strong three-valued logic. *Logical Investigations*, **20**, 253–266, 2014.
[74] A. Tamminga and K. Tanaka. A natural deduction system for first degree entailment. *Notre Dame Journal of Formal Logic*, **40**, 258–272, 1999.
[75] A. Urquhart. The complexity of decision procedures in relevance logic. In *Truth or Consequences: Essays in Honor of Nuel Belnap*, J. Dunn and A. Gupta, eds, pp. 61–76. Springer, Dordrecht, 1990.
[76] E. Voishvillo. Philosophical-methodological aspects of relevant logic. *Bulletin of the Section of Logic*, **18**, 146–152, 1989.

Received 5 January 2023