Towards Tractable Approximations to Many-Valued Logics: The Case of First Degree Entailment

MARCELLO D'AGOSTINO[1] AND ALEJANDRO SOLARES-ROJAS[2]

Abstract: **FDE** is a logic that captures relevant entailment between implication-free formulae and admits of an intuitive informational interpretation as a 4-valued logic in which "a computer *should* think". However, the logic is co-NP complete, and so an idealized model of how an agent *can* think. We address this issue by shifting to signed formulae where the signs express imprecise values associated with two distinct bipartitions of the set of standard 4 values. Thus, we present a proof system which consists of linear operational rules and only two branching structural rules, the latter expressing a generalized rule of bivalence. This system naturally leads to defining an infinite hierarchy of tractable depth-bounded approximations to **FDE**. Namely, approximations in which the number of nested applications of the two branching rules is bounded.

Keywords: **FDE**, tractability, natural deduction, tableaux

1 Introduction

Many interesting propositional logics are likely to be computationally intractable. For instance, Classical Propositional Logic (**CPL**) and First-Degree Entailment (**FDE**; Anderson & Belnap, 1962; Belnap, 1977a, 1977b; Dunn, 1976) are both co-NP complete (Arieli & Denecker, 2003; Cook, 1971; Urquhart, 1990). 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

[1]We thank two anonymous reviewers, Miguel Pérez-Gaspar and Pietro Casati for helpful comments. This research was funded by the Department of Philosophy "Piero Martinetti" of the University of Milan under the Project "Departments of Excellence 2018-2022" awarded by the Ministry of Education, University and Research (MIUR). The first author was also partly supported by MIUR under the PRIN 2017 project (n. 20173YP4N3).

[2]The second author was partly supported by the Israel Science Foundation (grant 550/19).

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., D'Agostino, 2015; D'Agostino, Finger, & Gabbay, 2013; D'Agostino & Floridi, 2009) 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 put forward by Quine (1973) to capture the "primitive meaning of the logical constants". The values have a natural informational interpretation ("accept", "reject", "abstain"). The proof-theoretic characterization given in (D'Agostino, 2015; D'Agostino et al., 2013) 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 she temporarily *assumes as if* she 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 point of view. In essence, each k-depth logic corresponds to a limited capability of manipulating virtual information.

The depth-bounded approach to **CPL**, as remarked in (D'Agostino, 2015), is the first stage 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 towards the first order case can be found in (D'Agostino, Larese, & Modgil, 2021). In this paper we show how the depth-bounded approach can be naturally extended to a useful many-valued system such as **FDE**. This provides a case study for extending the depth-bounded framework to a variety of finite-valued logics, in the spirit of (Caleiro, Marcos, & Volpe, 2015; Carnielli, 1987; Hähnle, 1999).

2 Belnap's semantics and the need for imprecise values

FDE arose out of the study of relevance logics.[3] Based on work of Dunn (1976), and an observation by Smiley (in correspondence), Belnap (1977a, 1977b) 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 Arieli, Avron, & Zamansky, 2018). An elegant 4-valued matrix is precisely Belnap-Dunn's.

In what follows we assume a standard propositional language $\mathcal{L}$ with $\wedge$, $\vee$ and $\neg$ as logical operators. We use $p, q, r, ...$, possibly with subscripts, as metalinguistic variables for atomic formulae; $A, B, C, ...$, possibly with subscripts, for arbitrary formulae; and $\Gamma, \Delta, \Lambda, ...$, possibly with subscripts, to vary over sets of formulae. Let $F(\mathcal{L})$ and $At(\mathcal{L})$ respectively be the set of well-formed formulae and atomic formulae of $\mathcal{L}$. Moreover, we define the *degree* of a formula A as the number of occurrences of connectives in A. 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 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 that p is true by some source and that p is false by some other source (or by the same source in different times); $\mathbf{n}$

[3] For a survey on **FDE** see (Omori & Wansing, 2017).

means that the computer is told nothing about the value of p. In essence, each value represents a subset of $\{\mathit{true}, \mathit{false}\}$ (Dunn, 1976). 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 $\mathbf{t}$ if p were previously marked with $\mathbf{n}$, with $\mathbf{b}$ if p were previously marked with $\mathbf{f}$; and of course leaves things alone if p was already marked either $\mathbf{t}$ or $\mathbf{b}$ (Belnap, 1977a, p. 12).

A *set-up* is simply an assignment to each of the atoms of exactly one of the values in $\mathbf{4}$. The values of complex formulae are obtained by means of considerations related to "Scott's thesis" about approximation lattices (Belnap, 1977a), resulting in the truth-tables in Table 1. Using these truth-tables, every set-up can be extended to a valuation function $v : F(\mathcal{L}) \longrightarrow \mathbf{4}$ in the usual inductive way. We call this function a $\mathbf{4}$-*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.[4] This is achieved by turning Belnap-Dunn's matrix into a valuation system where the set $\mathcal{V}$ of values is equal to $\mathbf{4}$, and the set $\mathcal{D}$ of designated values is equal to $\{\mathbf{t}, \mathbf{b}\}$. (Warning: do not confuse the values in **4** with *true* and *false*. The latter are *local* values referring to the information coming from a source, the former are *global* values, summarizing the epistemic state of the computer with respect to all the sources.) The consequence relation is then defined as follows:

Definition 1 $\Gamma \vDash A$ *iff for every* $\mathbf{4}$-*valuation* v, *if* $v(B) \in \{\mathbf{t}, \mathbf{b}\}$ *for all* $B \in \Gamma$, *then* $v(A) \in \{\mathbf{t}, \mathbf{b}\}$.

[4] As one anonymous reviewer pointed out to us, there is a tension between a justification in terms of information and the propositional attitude of knowledge in Belnap's seminal papers (Belnap, 1977a, 1977b), which is addressed in (Wansing & Belnap, 2010).

$\vee$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{b}$	$\mathbf{n}$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{b}$	$\mathbf{n}$
$\mathbf{b}$	$\mathbf{t}$	$\mathbf{b}$	$\mathbf{b}$	$\mathbf{t}$
$\mathbf{n}$	$\mathbf{t}$	$\mathbf{n}$	$\mathbf{t}$	$\mathbf{n}$

$\wedge$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{b}$	$\mathbf{n}$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{b}$	$\mathbf{n}$
$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
$\mathbf{b}$	$\mathbf{b}$	$\mathbf{f}$	$\mathbf{b}$	$\mathbf{f}$
$\mathbf{n}$	$\mathbf{n}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{n}$

$\neg$	
$\mathbf{t}$	$\mathbf{f}$
$\mathbf{f}$	$\mathbf{t}$
$\mathbf{b}$	$\mathbf{b}$
$\mathbf{n}$	$\mathbf{n}$

Table 1: **FDE**-tables

As is well-known, the relation $\vDash$ above is a (finitary) Tarskian consequence relation. That is, it satisfies the following conditions:

Reflexivity: If $A \in \Gamma$, then $\Gamma \vDash A$.

Monotonicity: If $\Gamma \vDash A$, then $\Gamma \cup \Delta \vDash A$.

Cut: If $\Gamma \vDash A$ and $\Gamma \cup \{A\} \vDash B$, then $\Gamma \vDash B$.

Further, for the unrestricted language allowing arbitrary formulae involving $\wedge, \vee$ and $\neg$, the decision problem for this consequence relation is co-NP complete (see, Arieli & Denecker, 2003; Urquhart, 1990), which 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 implicit in the quotation from Belnap above, the values in $\mathbf{4}$, except for $\mathbf{b}$, 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 $\mathbf{t}$, $\mathbf{f}$, $\mathbf{n}$ as stable we need to assume complete information about the set of sources Ω. Namely, while the meaning of $\mathbf{b}$ 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 $\mathbf{t}$, $\mathbf{f}$ and $\mathbf{n}$ 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 $\mathbf{4}$ are *information-transcendent* when interpreted as timeless. They refer to an objective state of affairs 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 "$\mathbf{t}$ *or* $\mathbf{b}$", which is implicit in the choice of the set of designated values by Belnap. Inspired by (D'Agostino, 1990) and (Avron, 2003; Fitting, 1991, 1994), we shall address this question by shifting to *signed formulae*, where the signs express such imprecise values associated with two distinct bipartitions of $\mathbf{4}$.

A *second* key observation is that, as suggested by Belnap (1977a, 1977b), 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 which was anticipated by Quine. (See D'Agostino (2014) for further references and a discussion that includes an interesting quotation from Michael Dummett to the effect that in non-mathematical contexts our information may well be *irremediably disjunctive* in nature.)

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 (D'Agostino, 2015; D'Agostino et al., 2013; D'Agostino & Floridi, 2009) for **CPL**. Before addressing this issue, however, we shall provide in the next section a proof-theoretic characterization of depth-unbounded reasoning in **FDE** that will pave the way for defining its tractable approximations.

3 Intelim deduction in FDE

Signed formulae (S-formulae, for short) are expressions of the form $\mathsf{T}\,A$, $\mathsf{F}\,A$, $\mathsf{T}^*\,A$ and $\mathsf{F}^*\,A$. 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" ($v(A) \in \{\mathbf{f}, \mathbf{b}\}$).[5] Crucially, S-formulae of the form $\mathsf{T}\,A$ or $\mathsf{F}^*\,A$ express information that x may hold even without a complete knowledge of the set of 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 $\mathsf{T}^*\,A$ is $\mathsf{F}^*\,A$ and vice versa. Besides, we write $\mathsf{T}\,\Gamma$ for $\{\mathsf{T}\,A \mid A \in \Gamma\}$. Moreover, we use $\varphi, \psi, \theta, ...$, possibly with subscripts, as variables ranging over S-formulae; and $X, Y, Z, ...$, 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 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" state of affairs concerning the whole of Ω. In other words, without making hypothetical assumptions that go beyond the information that she 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 (D'Agostino, 2015; D'Agostino et al., 2013) 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

[5] Similar approaches to **FDE** are given in (Blasio, 2015, 2017) and (Shramko & Wansing, 2005), 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 the latter approach.

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

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

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

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

Table 2: Introduction rules for the standard **FDE** connectives

of the basic logic bears some resemblance with natural deduction, but does not have discharge rules, since no hypothetical reasoning is involved. 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 $(\varphi_1, ..., \varphi_n)$ s.t., for every $i = 0, ..., n$, either φ_i is an assumption or it is the conclusion of the application of an intelim rule to preceding S-formulae.

The intelim rules are all sound, but not complete for full **FDE**. Indeed, as we shall show, these rules just characterize the basic logic in the hierarchy. Completeness 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. Their intuitive meaning 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 related to a *generalized* Principle of Bivalence:[6]

[6]Generalizations of the rule of bivalence have been fruitfully used in the context of many-valued and substructural logics (see Caleiro et al., 2015; D'Agostino, Gabbay, & Broda, 1999; Hähnle, 1999).

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

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

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

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

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

Table 3: Elimination rules for the standard **FDE** connectives

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

For **CPL** only the first rule, with T and F interpreted as “only true” and “only false”, makes sense and is sufficient for completeness. With the addition of these rules, deductions are represented by downward-growing trees, which brings the method somewhat closer to tableaux.[7] Each application of PB or PB^* stands for the introduction of virtual information about the imprecise value of a formula A, which we shall respectively call the *PB-formula* or *PB^*-formula*. Note once again that, whereas signed formulae of the form $\mathsf{T}\,A$ and $\mathsf{F}^*\,A$ are empirically obtainable, signed formulae of the form $\mathsf{T}^*\,A$ and $\mathsf{F}\,A$ express hypotheses introduced by applications of PB or PB^*. In turn, the S-formulae $\mathsf{T}\,A$, $\mathsf{F}\,A$, $\mathsf{T}^*\,A$ and $\mathsf{F}^*\,A$ appended via those branching rules will be all called *virtual assumptions*. Now, PB and PB^* are essentially cut rules that may introduce formulae of arbitrary degree. However, as we will show in Lemma 2, their application can be restricted so as to satisfy the subformula property. Moreover, from our informational viewpoint, the

[7] Well-known (Smullyan-style) tableaux for **FDE** were introduced by Priest (2001). Space prevent us to include a fair comparison of our investigation in this paper with related work; we shall include it in a subsequent paper.

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. 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 (D'Agostino, 1990) 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 implies the subformula property. In this paper we choose to prove a more general version of the subformula property by means of proof transformations.

Definition 2 *Let $X = \{\varphi_1, ..., \varphi_m\}$. Then $\mathcal{T}$ is an* intelim tree for X *if there is a finite sequence $(\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n)$ s.t. $\mathcal{T}_1$ is a one-branch tree consisting of any sequence of the formulae in X, $\mathcal{T}_n = \mathcal{T}$, 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 s.t. 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 is, of course, a kind of explosivity; but it regards *signed* formulae, and it 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:

Proposition 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. □

4 Subformula property

A proof has the *subformula property* if every formula in it is a subformula either of the assumptions or of the conclusion. In the case of refutations only subformulae of the assumptions can occur.

Definition 3 *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.*

Definition 4 *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: (i) it contains no idle occurrences of S-formulae; (ii) none of its branches contains more than one occurrence of the same S-formula; (iii) none of its branches properly includes a closed path.*

Lemma 1 *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. □

We observe that turning an intelim proof or refutation $\mathcal{T}$ into a non-redundant one is computationally easy. Now, let us denote by $\mathsf{sub}(\Delta)$ the closure under subformulae of the set Δ of formulae. Given a proof $\mathcal{T}$ of φ from X (a refutation of X) we say that an application of PB or PB^* in $\mathcal{T}$ is *analytic* if its PB-formula or PB^*-formula is in $\mathsf{sub}(X^u \cup \{\varphi^u\})$ ($\mathsf{sub}(X^u)$). We also say that an intelim tree is *analytic* if all the applications of PB and PB^* in it are analytic.

Lemma 2 *Every intelim proof* $\mathcal{T}$ *of* φ *from* X *(refutation of* X*) can be transformed into a proof* $\mathcal{T}'$ *of* φ *from* X *(refutation* $\mathcal{T}'$ *of* X*) s.t. every application of* PB *and* PB^* *in* $\mathcal{T}$ *is analytic.*

Proof. We use the notation $\begin{array}{c}\mathcal{T}\\ \varphi\end{array}$ to denote either the empty intelim tree or a non-empty intelim tree s.t. φ 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 analytic, and $\kappa(\mathcal{T})$ denotes the number of occurrences of such non- 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 a 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:

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

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

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

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

Given Lemma 1 and Lemma 2, it is not difficult to show that:

Theorem 1 (Subformula property) *If $\mathcal{T}$ is a proof of φ from X (a refutation of X), $\mathcal{T}$ can be transformed into a proof (refutation) $\mathcal{T}'$ of φ from X (of X) with the subformula property.*

5 Depth-bounded approximations to FDE

Definition 5 *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 its depth is* k*.*

Definition 6 *For all* X*,* φ*, (i)* φ *is* 0-depth deducible from X, $X \vdash_0 \varphi$*, iff there is a 0-depth intelim proof of* φ *from* X*; (ii)* X *is* 0-depth refutable*,* $X \vdash_0$*, iff there is a 0-depth intelim refutation of* X*.*

We shall abuse of the same relation symbol '$\vdash_0$' to denote 0-depth deducibility and refutability. It is easy to show that:

Proposition 2 $\vdash_0$ *is a Tarskian consequence relation.*

Definition 7 *For all* X*,* φ*, and for* $k > 0$*,*

$X \vdash_k \varphi$ *iff* $X \cup \{\psi\} \vdash_{k-1} \varphi$ *and* $X \cup \{\bar{\psi}\} \vdash_{k-1} \varphi$ *for some* $\psi^u \in \mathsf{sub}(X^u \cup \{\varphi^u\})$.

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

We shall abuse of the same relation symbol '$\vdash_k$' to denote k-depth deducibility and refutability. Now, it follows immediately from Definitions 5 and 7 that:

Proposition 3 *For all* X*,* φ*,* $X \vdash_k \varphi$ *(*$X \vdash_k$*) iff there is a* k*-depth proof of* φ *from* X *(a* k*-depth refutation of* X*) s.t. all its* PB*-formulae and* PB^**-formulae are in* $\mathsf{sub}(X^u \cup \{\varphi^u\})$ *(*$\mathsf{sub}(X^u)$*).*

As is the case for **CPL**:

Proposition 4 *The* k*-depth deducibility relations* $\vdash_k$ *satisfy reflexivity, monotonicity, but not cut.*

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

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

Further, since $\vdash_0$ is monotonic, its successors are ordered: $\vdash_j \subseteq \vdash_k$ whenever $j \leq k$. The transition from $\vdash_k$ to $\vdash_{k+1}$ corresponds to an increase in the depth at which the nested use of virtual information is allowed. From the adequacy of the unbounded system and the subformula property, it immediately follows that

Proposition 5 *$X \vdash \varphi$ in* **FDE** *iff $X \vdash_k \varphi$ for some k.*

We conclude by observing that the decision problem for the k-depth logics is tractable. Theorem 1 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 subformula property. 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, $k > 0$.

Theorem 2 *Whether or not $X \vdash_0 \varphi$ ($X \vdash_0$) can be decided in time $O(n^2)$, where n is the total number of occurrences of symbols in the elements of $X \cup \{\varphi\}$ (of X).*

Proof sketch. The proof can be adapted from (D'Agostino et al., 2013). We just sketch the decision procedure and give a hint about the upper bound.

We now describe a general procedure to generate the closure of a set Y of signed formulae under the intelim rules restricting our attention to a finite search space Δ that includes all the formulae in Y^u and is closed under subformulae. Start by constructing the subformula graph associated with Δ, i.e., the graph in which the nodes are the subformulae of Δ, while the edges represent the subformula relation. Observe that the number of distinct subformulae of a formula is always less than or equal to the number of occurrences of symbols in that formula. So the number of distinct subformulae of the formulae in Y is $O(n)$, where n is the number of occurrences of symbols in the elements of Y. Constructing this graph takes time $O(n^2)$. A *neighbour* of a node A is a node consisting of either (i) one of the immediate subformulae of A (if any), or (ii) one of the immediate superformulae of A (if any), or (iii) else one of the immediate subformulae of the immediate superformulae of A (if any). The number of neighbours of each node is $O(n)$.

Let us say that a node in the subformula graph is associated with a premise (a conclusion) of an intelim rule, if it consists of a formula that is the unsigned part of a premise or of the conclusion. Note that

The relation "A is a neighbour of B" is symmetric. (1)

The node associated with a premise of an intelim rule is a neighbour both of the node associated with the second premise (if any) and of the node associated with the conclusion. (2)

Nodes are labelled with a subset of the four signs as follows. Initially, all nodes are marked as "fulfilled". Whenever a new sign is added to the labelling set, the node turns "unfulfilled". At the beginning all the nodes consisting of the formulae in X^u are labelled in accordance with their signs in X (and therefore turn "unfulfilled") while all the others are labelled with the empty set. Fulfilling a node means that all the possible intelim rules involving this node and any of its neighbours are applied, which may lead to adding new signs to the labelling sets of the nodes in the neighbours, making them unfulfilled. This amounts to using the formula in the node to be fulfilled, prefixed with each of the signs in its labelling set, as premise of an intelim rule, possibly involving one of its neighbours as second premise. Yet-unfulfilled nodes are fulfilled in turn (the order is immaterial) and marked as such. Since there are $O(n)$ neighbours, fulfilling a node takes $O(n)$ steps.

A node is *inconsistent* if its label contains a pair of conjugate signs, otherwise it is *consistent*. Note that the labelling set of each consistent node may contain at most two signs. Note also that it may be necessary to fulfil a node more than once, when a *new* sign is added to its labelling set as a result of an application of an intelim rule to one of its neighbours. However, no consistent node needs to be fulfilled more than twice (once for each sign in its labelling set). To see this, observe that if the procedure leads to adding a new sign to a node n' (e.g., the sign F to A) that may, in turn, be used together with a previously fulfilled neighbour n (e.g., $A \vee B$ signed with T) as premise of an intelim rule, then n' turns unfulfilled, and the rule in question will be applied anyway when fulfilling n'. For, n is a neighbour of n', by (1), and so is the node n'' consisting of the conclusion of the rule application (B), by (2), whose labelling set will be updated accordingly (adding T).

A *graph* is *inconsistent* if it contains an inconsistent node, otherwise it is *consistent*. In turn, a graph is 0-*depth saturated* if it is either inconsistent or it is consistent and each node is marked as fulfilled. A 0-depth saturated graph is obtained in $O(n^2)$ steps, since there are $O(n)$ nodes in the graph, each node is fulfilled at most twice and fulfilling a node takes $O(n)$ steps. Figure 1 shows the initialized graph for the set $X = \{\mathsf{T}\, C \vee (A \vee B), \mathsf{F}\, C, \mathsf{F}\, A \vee (B \wedge \neg C)\}$.

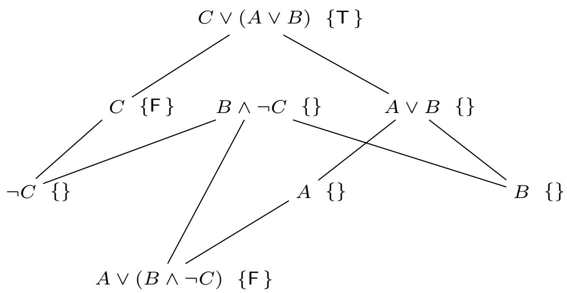

Figure 1: Initialized graph

The corresponding saturated graph, with a possible order of fulfillment of the nodes, is shown in Figure 2. The reader can verify that any alternative sequence leads to the same saturated graph. Figure 3 shows the initialized graph for the set $X = \{\mathsf{T}\, A \vee (B \wedge C), \mathsf{F}\, (A \vee B) \wedge (A \vee C), \mathsf{F}\, A\}$. A corresponding saturated graph, with a possible order of fulfillment of the nodes, is shown in Figure 4. Note that, for inconsistent graphs, not any alternative sequence leads to the same saturated graph. In general, all the signed formulae φ of the form $\mathsf{S}\, A$, where S is in the labelling set of A, that occur in a saturated graph are 0-depth deducible from X.

To decide whether $X \vdash_0 \varphi$ ($X \vdash_0$), consider the graph associated with $X^u \cup \varphi^u$ (X^u), initialize it by adding signs to the labelling sets in accordance with X, and then run the saturation procedure. When the graph is saturated $X \vdash_0 \varphi$ iff the sign of the signed formula φ belongs to the labelling set of φ^u or the graph is inconsistent. Note that an inconsistent graph detects a "metalevel" inconsistency that concerns an incoherent assignment of the imprecise values associated with the signs. Note also that a 0-depth saturated graph starting with nodes labelled with $\{\mathsf{T}\}$ is always consistent and may contain only the signs T and F^* in the labelling sets. □

Corollary 1 *Whether or not $X \vdash_k \varphi$ ($X \vdash_k$) can be decided in time $O(n^{k+2})$, where n is the total number of occurrences of symbols in the elements of $X \cup \{\varphi\}$ (of X).*

Hint. From Definition 7 and the observation that there are $O(n)$ distinct subformulae of $X^u \cup \{\varphi^u\}$ (X^u).

□

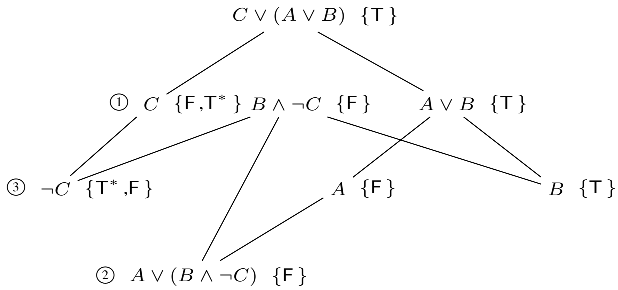

Figure 2: Saturated graph

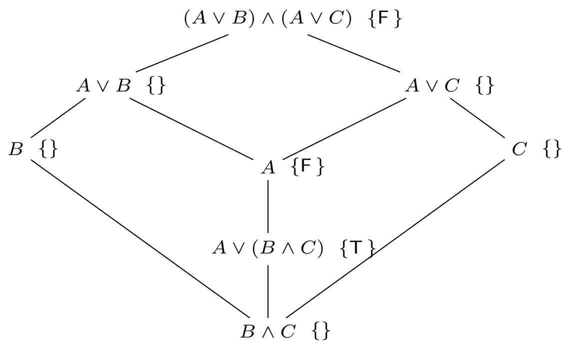

Figure 3: Initialized graph

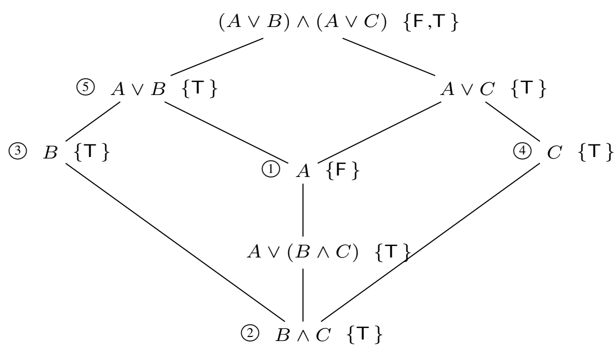

Figure 4: Saturated graph

6 Conclusions

We approached **FDE** via a proof system formulated by means of signed formulae, where the signs express imprecise values associated with two distinct bipartitions of the set of standard 4 values. The proof system consists of linear intelim rules and two branching rules expressing a generalized rule of bivalence, and naturally leads to defining an infinite hierarchy of tractable approximations to **FDE** by controlling the application of the latter rules. Unlike the intelim rules, the branching rules introduce hypothetical information about the imprecise value of a formula. Intuitively, the more virtual information needs to be invoked via the branching rules, the harder the inference is for an agent. So, the nested application of those rules provides a sensible measure of inferential depth.

In a subsequent paper we shall show that our hierarchy of approximations to **FDE** admits of an intuitive 5-valued non-deterministic semantics. This semantics essentially takes the signs as imprecise values (i.e., two-element sets of the standard values), and a fifth value is introduced to represent the case where the agent's information is insufficient to even establish any of the imprecise values. Further, following the methodology used in this paper, we shall define analogous hierarchies of depth-bounded approximations to two logics closely related to **FDE**, namely, Priest's Logic of Paradox and the Strong Kleene Logic.

References

Anderson, A., & Belnap, N. (1962). Tautological entailments. *Philosophical Studies*, *13*(1-2), 9–24.

Arieli, O., Avron, A., & Zamansky, A. (2018). *Theory of Effective Propositional Paraconsistent Logics*. London: College Publications.

Arieli, O., & Denecker, M. (2003). Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, *13*(4), 557–580.

Avron, A. (2003). Tableaux with four signs as a unified framework. In M. Cialdea Mayer & F. Pirri (Eds.), *Automated Reasoning with Analytic Tableaux and Related Methods, Int. Conf., TABLEAUX 2003* (Vol. 2796, pp. 4–16). Berlin, Heidelberg: Springer.

Belnap, N. (1977a). How a computer should think. In G. Ryle (Ed.), *Contemporary Aspects of Philosophy* (pp. 30–55). Stocksfield: Oriel Press.

Belnap, N. (1977b). A useful four-valued logic. In J. M. Dunn & G. Epstein (Eds.), *Modern Uses of Multiple-Valued Logics* (pp. 5–37). Dordrecht: Reidel Publishing Company.

Blasio, C. (2015). 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).

Blasio, C. (2017). Revisitando a lógica de Dunn-Belnap. *Manuscrito*, *40*(2), 99–126.

Caleiro, C., Marcos, J., & Volpe, M. (2015). Bivalent semantics, generalized compositionality and analytic classic-like tableaux for finite-valued logics. *Theoretical Computer Science*, *603*, 84–110.

Carnielli, W. (1987). Systematization of finite many-valued logics through the method of tableaux. *The Journal of Symbolic Logic*, *52*(2), 473–493.

Cook, S. (1971). The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (pp. 151–158). New York, NY: Association for Computing Machinery.

D'Agostino, M. (1990). *Investigations into the Complexity of some Propositional Calculi*. Oxford University. Computing Laboratory. Programming Research Group.

D'Agostino, M. (2014). Analytic inference and the informational meaning of the logical operators. *Logique et Analyse*, *57*(227), 407–437.

D'Agostino, M. (2015). An informational view of classical logic. *Theoretical Computer Science*, *606*, 79–97.

D'Agostino, M., Finger, M., & Gabbay, D. (2013). Semantics and proof-theory of depth bounded Boolean logics. *Theoretical Computer Science*, *480*, 43–68.

D'Agostino, M., & Floridi, L. (2009). The enduring scandal of deduction. *Synthese*, *167*(2), 271–315.

D'Agostino, M., Gabbay, D., & Broda, K. (1999). Tableau methods for substructural logics. In M. D'Agostino, D. M. Gabbay, R. Hähnle, & J. Posegga (Eds.), *Handbook of Tableau Methods* (pp. 397–467). Dordrecht: Springer.

D'Agostino, M., Larese, C., & Modgil, S. (2021). Towards depth-bounded natural deduction for classical first-order logic. *Journal of Applied Logics*, *8*(2), 423–451.

Dunn, J. (1976). Intuitive semantics for first-degree entailments and 'coupled trees'. *Philosophical Studies*, *29*(3), 149–168.
Fitting, M. (1991). Bilattices and the semantics of logic programming. *The Journal of Logic Programming*, *11*(2), 91–116.
Fitting, M. (1994). Kleene's three valued logics and their children. *Fundamenta Informaticae*, *20*(1, 2, 3), 113–131.
Hähnle, R. (1999). Tableaux for many-valued logics. In M. D'Agostino, D. M. Gabbay, R. Hähnle, & J. Posegga (Eds.), *Handbook of Tableau Methods* (pp. 529–580). Dordrecht: Springer.
Omori, H., & Wansing, H. (2017). 40 years of FDE: An introductory overview. *Studia Logica*, *105*(6), 1021–1049.
Priest, G. (2001). *An Introduction to Non-Classical Logic*. Cambridge: Cambridge University Press.
Quine, W. (1973). *The Roots of Reference*. LaSalle: Open Court.
Shramko, Y., & Wansing, H. (2005). Some useful 16-valued logics: How a computer network should think. *Journal of Philosophical Logic*, *34*(2), 121–153.
Urquhart, A. (1990). The complexity of decision procedures in relevance logic. In J. M. Dunn & A. Gupta (Eds.), *Truth or Consequences: Essays in Honor of Nuel Belnap* (pp. 61–76). Dordrecht: Springer Netherlands.
Wansing, H., & Belnap, N. (2010). Generalized truth values. A reply to Dubois. *Logic Journal of IGPL*, *18*(6), 921–935.

Marcello D'Agostino
University of Milan, Department of Philosophy
Italy
E-mail: `marcello.dagostino@unimi.it`

Alejandro Solares-Rojas
University of Milan, Department of Philosophy
Italy
E-mail: `alejandro.solares@unimi.it`