An Informational View of Classical Logic☆

Marcello D'Agostino

Department of Economics and Management, University of Ferrara, Ferrara, Italy

This is a post-print version, formatted by the author, of the paper:
M. D'Agostino, An Informational View of Classical Logic, Theoret. Comput. Sci. 606:79-97 (2015), http://dx.doi.org/10.1016/j.tcs.2015.06.057.

Abstract

We present an informational view of classical propositional logic that stems from a kind of informational semantics whereby the meaning of a logical operator is specified solely in terms of the information that is *actually* possessed by an agent. In this view the inferential power of logical agents is naturally bounded by their limited capability of manipulating "virtual information", namely information that is not implicitly contained in the data. Although this informational semantics cannot be expressed by any finitely-valued matrix, it can be expressed by a non-deterministic 3-valued matrix that was first introduced by W.V.O. Quine, but ignored by the logical community. Within the general framework presented in [21] we provide an in-depth discussion of this informational semantics and a detailed analysis of a specific infinite hierarchy of tractable approximations to classical propositional logic that is based on it. This hierarchy can be used to model the inferential power of resource-bounded agents and admits of a uniform proof-theoretical characterization that is half-way between a classical version of Natural Deduction and the method of semantic tableaux.

Keywords: Classical Propositional Logic, Informational Semantics, Non-deterministic matrices, Computational Complexity, Natural Deduction sep Semantic Tableaux.

☆This paper elaborates on ideas and results arising from my collaboration with Marcelo Finger, Luciano Floridi and Dov Gabbay [22, 21] and is based on an invited talk given at the 8th Workshop on Logical and Semantic Frameworks with Applications (LSFA) held in São Paulo on 2–3 September 2013. A preliminary and partial version was published in the proceedings of the workshop [20]. I wish to thank Maribel Fernandez and Marcelo Finger for inviting me to the workshop and all the participants for an interesting discussion. I also wish to thanks two anonymous referees for valuable comments and suggestions.

Email address: dgm@unife.it (Marcello D'Agostino)

1. The background problem

The fundamental question that we address in this paper is the following:

do we *actually* possess the information that the conclusion of an inference is true *whenever* we possess the information that its premises are true? (1)

The lack of a general decision procedure strongly suggests that the intuitive answer is "no" in the domain of classical first-order logic: there is no guarantee that we are in a position to *effectively* recognize the truth of a valid consequence A of a set Γ of sentences in all informational situations in which we recognize the truth of the sentences in Γ. Moreover, despite the existence of decision procedures for classical propositional logic, the widely believed conjecture that $P \neq NP$, makes it highly improbable that there exists a feasible one. So, again, there is no guarantee that we are in a position to *feasibly* recognize that the conclusion of a valid propositional inference is true in all informational situations in which we recognize that its premises are true. Therefore, if we construe the notion of "actually possessing" a piece of information as having access to it in practice,[1] and not only in principle, a positive answer to (1) sounds highly counterintuitive even in the restricted domain of propositional logic.

In fact, standard logical systems provide adequate models of *logically omniscient* agents, a normative ideal that can only be approximated in practice. This is a source of major difficulties in all research areas where there is an urgent need for less idealized, yet theoretically principled, models of logical agents with bounded cognitive and computational resources. From this point of view, it makes sense to require as in [36] that a logical system should consist not only in an algorithmic or semantic characterization of a logic L, but also in a definition of how this logic L can be *approximated in practice* by realistic agents, no matter whether human or artificial.

Despite various interesting, albeit scattered and differently motivated, contributions,[2] logic still lacks solid general foundations for an approximation theory. In this paper we elaborate on ideas and results presented in a series of previous papers [16, 22, 17, 21, 18, 19] to make a step in this direction by outlining an "informational view" of classical propositional logic that naturally yields a sequence of tractable approximations. We start from the following problem: for which subsystems of classical propositional logic does it make intuitive sense to give a positive answer to (1)? As argued in [21], a rather natural solution stems from an alternative informational semantics for the Boolean operators, whereby the meaning of a logical operator is specified solely in terms of the information that is *actually possessed* by an agent. This semantics leads

[1] This may include implicit information that can be extracted from the information explicitly held by an agent (whether human or artificial) via a natural feasible procedure, i.e., one that is part of the agent's "built-in" inferential procedures.

[2] With no claim of exhaustivity, we mention [10, 47, 27, 28, 13, 43, 48, 31, 32, 34, 35, 33, 39].

to an incremental characterization of Boolean logic as the limit of a sequence of tractable depth-bounded subsystems of increasing inferential power (and increasing computational complexity).[3]

While the semantic definition of this hierarchy is independent of any specific proof-theoretical formalism, it admits of a simple characterization in terms of a proof system that is half-way between a classical version of Natural Deduction and the method of semantic tableaux. The basic 0-depth logic is naturally characterized by means of a set of introduction and elimination rules. These can be seen as natural deduction rules that are logically weaker than the standard Gentzen-style rules, in that they involve no "discharge" of assumptions, and are more suitable to represent the classical meaning of the logical operators. Alternatively, they can also be seen as a kind of tableau-like rules that extend the elimination rules of the KE system [25] via a set of introduction rules.

The increasing inferential power of each k-depth approximation (with $k > 0$) depends only on a single *structural rule* and on the depth at which its application is allowed. This structural rule is, in essence, a (classical) *cut* rule — closely related to the Principle of Bivalence — that governs the manipulation of "virtual information", i.e., information that we do not actually possess, but we temporarily assume *as if* we possessed it. In our approach, therefore, the answer to (1) is a *matter of degree* and depends on the minimum depth at which the use of virtual information is required to obtain the conclusion from the premises.

The main new contributions of this paper with respect to [21] are the following: (i) we focus on a specific hierarchy of depth-bounded approximations to Boolean Logic belonging to one of the families abstractly discussed in [21] and present in more detail their semantic and proof-theoretical properties, with clarifying examples; (ii) we provide an in-depth discussion of an *intuitive* informational semantics for the basic (0-depth) system of this hierarchy that was anticipated back in the 1970's by some observations of Willard V.O. Quine [46] (with no connection with tractable inference) and can be expressed by a 3-valued non-deterministic matrix; this semantics was subsequently and independently re-proposed (with no apparent connection with the intuitive interpretation given by Quine) by Crawford and Etherington [13] who claimed (without proof) that it provides a characterization of unit resolution;[4] here we support their intuition that this semantics may become the basic foundational tool for a general theory of tractable approximations to classical logic, by showing that it captures exactly all the logical inferences that can be drawn by using only "actual information"; we also show that its scope is much wider than what envisaged in [13],

[3]The hierarchy discussed in this paper originates from a long-standing research program started in [14] and can be interestingly compared to the similar hierarchies proposed in the contributions cited in Footnote 2. From the proof-theoretical viewpoint, it is related to Stålmarck's method [51, 48, 8], which has proved quite successful in the area of system verification (see [9]); for a more detailed discussion of this relation see [21].

[4]This non-deterministic semantics is briefly discussed also in [21] and claimed to be equivalent to the "modular" semantics thoroughly investigated in that paper.

in that it is relevant to *any* logical formalism with no syntactic restrictions;[5] (iii) we provide a direct proof of the completeness of classical 0-depth deduction with respect to this 3-valued non-deterministic semantics; (iv) we provide a detailed discussion of *depth-bounded intelim trees* — a natural proof-theoretical characterization of the hierarchy defined via the informational semantics — and a very simple proof of a normal form theorem that is more general and more informative than the subformula theorem in [21].

2. An informational semantics for the Boolean operators

The classical meaning of the logical operators is usually specified by the familiar truth-tables that fix the conditions under which a sentence is true or false in terms of the truth or falsity of its immediate constituents. The underlying notions of truth and falsity are assumed to obey the two classical principles of Bivalence (any sentence is either true or false independent of our holding any information about it) and Non-Contradiction (no sentence can be at the same time true and false). This way of fixing the meaning of a logical operator is perfectly in tune with the classical, information-transcendent, notions of truth and falsity and with the traditional view of logical inference as a truth-transmission device; but it is at odds with the equally important view of logical inference as an information-processing device. To abide by the latter view we need a semantics based on informational notions. Moreover, in order to define subsystems of classical logic that justify a positive answer to (1), we need a semantics based on the notion of *actual information*, i.e., to put it with Jaakko Hintikka, information that "we actually possess (as distinguished from the information we in some sense have potentially available to us) and with which we can in fact operate" [38, p. 229].

The primary notions of this semantics, therefore, are not classical truth and falsity, but *informational truth* and *informational falsity*, namely *holding the information* that a sentence is true, respectively false. Here, by saying that an agent x *holds* the information that A is true (respectively false) we mean that this is information that is *practically available* to x and with which x can operate. Clearly, these notions do not satisfy the informational version of the Principle of Bivalence: it may well be that for a given A, we neither hold the information that A is true, nor do we hold the information that A is false. On the other hand, in this paper we assume that they do satisfy the informational version of the Principle of Non-Contradiction: no agent can *actually* possess both the information that A is true and the information that A is false, as this

[5] In fact, the discussion of the 0-depth case in [13] is restricted to formulae in negation normal form. This requires preliminary translation via the De Morgan laws. However, the equivalence $\neg(A \wedge B) \equiv \neg A \vee \neg B$ is not sound under the non-deterministic semantics, and this generates unnecessary anomalies such as the failure of modus ponens (see Example 2). Moreover, the k-depth semantics presented in this paper is based on a classical *reductio ad absurdum* that applies only to formulae in clausal form.

would be deemed to be equivalent to possessing no definite information about A.[6]

We use the values 1 and 0 to represent, respectively, informational truth and falsity. When a sentence takes neither of these two defined values, we say that it is *informationally indeterminate*. It is technically convenient to treat informational indeterminacy as a third value that we denote by "$\perp$".[7] The three values are partially ordered by the relation $\preceq$ such that $v \preceq w$ ("v is less defined than, or equal to, w") if, and only if, $v = \perp$ or $v = w$ for $v, w \in \{0, 1, \perp\}$.

Note that the old familiar truth tables for $\wedge, \vee$ and $\neg$ are still intuitively sound under this informational reinterpretation of 1 and 0. For example, if we hold the information that A is true and the information that B is true, then we thereby hold the information that $A \wedge B$ is true, etc. However, they are no longer exhaustive: they do not tell us what happens when one or all of the immediate constituents of a complex sentence take the value $\perp$.

So, we need to conservatively extend the classical truth-tables with new entries to accommodate the third value $\perp$. More precisely, for every n-ary Boolean operator $\star$, whose classical meaning is fixed by a truth-function $f_\star$, we want to specify its informational meaning as given by some sort of function $\hat{f}_\star$ satisfying:

$$\hat{f}_\star(z_1, \ldots, z_n) = f_\star(z_1, \ldots, z_n), \text{ whenever } z_1, \ldots, z_n \in \{0, 1\}. \tag{2}$$

Given our interpretation of the third value $\perp$ as informational indeterminacy, a reasonable requirement is also that our logical operators are *monotonic* in the following sense:

$$v_1 \preceq w_1 \text{ and } \ldots \text{ and } v_n \preceq w_n \Longrightarrow \hat{f}_\star(v_1, \ldots, v_n) \preceq \hat{f}_\star(w_1, \ldots, w_n) \tag{3}$$

Let us, from now on, restrict our attention to the logical operators $\wedge, \vee, \neg$. Under the requirements (2) and (3), the tables of Kleene's (strong) 3-valued logic [40, §64], shown in Table 1, may appear as the most natural candidates to represent their informational meaning. However, while the table for negation appears perfectly in tune with our informational interpretation of the three values, the tables for $\wedge$ and $\vee$ are not, in that they do not appear to account for some of our intuitive judgments. Typical counterevidence is presented in the following quotations from Willard V.O. Quine taken from his book *The Roots of Reference* and concerning what he there calls "the primitive meaning of the logical operators". This is expressed in terms of an agent's *disposition* to assent or dissent to a sentence in a given informational situation:

[6] Notice that this assumption does not rule out the possibility of *hidden* inconsistencies in an agent's information state, but only of inconsistencies that can be feasibly detected by that agent. It is, however, possible to investigate paraconsistent variants of the semantics proposed in this paper in which even this weak informational version of the Principle of Non-Contradiction is relaxed. This will be the subject of a subsequent paper.

[7] This is the symbol for "undefined", the bottom element of the information ordering, not to be confused with the "falsum" logical constant that we shall denote by "$\curlywedge$".

$\wedge$	1	0	$\perp$
1	1	0	$\perp$
0	0	0	0
$\perp$	$\perp$	0	$\perp$

$\vee$	1	0	$\perp$
1	1	1	1
0	1	0	$\perp$
$\perp$	1	$\perp$	$\perp$

$\neg$	
1	0
0	1
$\perp$	$\perp$

Table 1: Kleene's 3-valued tables.

> Conjunction has its blind spot [...] when neither component commands assent or dissent. There is no direct way of mastering this quarter. In some such cases the conjunction commands dissent and in others it commands nothing. This sector is mastered only later, in theory-laden ways. Where the components are "it is a mouse" and "it is a chipmunk", and neither is affirmed nor denied, the conjunction will still be denied. But where the components are "it is a mouse" and "it is in the kitchen", and neither is affirmed nor denied, the conjunction will perhaps be left in abeyance.
> [...]
> Alternation, like conjunction, has its blind quarter where neither component commands assent or dissent. We might assent to the alternation of "it is a mouse" and "it is chipmunk" or we might abstain [46, p. 77].

In general, when we are faced with a conjunction $A \wedge B$ in which both A and B are informationally indeterminate, the value of the conjunction may be either informational falsity 0, or informational indeterminacy $\perp$, depending on whether or not we hold the additional information that A and B cannot be simultaneously true. And the value of $A \vee B$ may be either informational truth 1 or informational indeterminacy $\perp$, depending on whether or not we hold the additional information that at least one of A and B must be true.[8] This discussion strongly suggests that Kleene's 3-valued tables are not apt to capture the informational meaning of the logical operators $\vee$ and $\wedge$ and that, indeed, no system of standard *deterministic* tables can do any better. Quine's suggestion, reported in the above quotations, leads to the following *non-deterministic* tables for $\wedge$ and $\vee$:

$\wedge$	1	0	$\perp$
1	1	0	$\perp$
0	0	0	0
$\perp$	$\perp$	0	$\perp, 0$

$\vee$	1	0	$\perp$
1	1	1	1
0	1	0	$\perp$
$\perp$	1	$\perp$	$\perp, 1$

Here, the entries in which both arguments are $\perp$ yield two alternative possi-

[8] As far as the operator $\vee$ is concerned, its informational meaning we are discussing here clearly departs from its intuitionistic meaning, according to which a disjunction $A \vee B$ is intuitionistically true (roughly speaking, provable) if and only if either A is intuitionistically true or B is intuitionistically true. This is the so-called disjunction property of intuitionistic logic. While this property is appropriate for (constructive) mathematics, it is quite at odds with ordinary usage outside mathematics. On this point see [29], pp. 266–267 and 277–278.

ble values, meaning that the value of the compound sentence is not uniquely determined by the values of its immediate constituents, but can be either of the two values shown. In other words, the "function" $\hat{f}_\star$ that fixes the informational meaning of a binary operator $\star$ is a *non-deterministic truth-function*.[9] These non-deterministic tables where independently rediscovered by Crawford and Etherington [13] who claimed that they provide a semantic characterization of unit-resolution. The general theory of non-deterministic matrices has been brought to the attention of the logical community and extensively investigated by Arnon Avron and co-authors (see [3, 4, 1, 2, 5] among others).

A non-deterministic table for the informational meaning of the Boolean conditional can be obtained in the obvious way:

$\rightarrow$	1	0	$\perp$
1	1	0	$\perp$
0	1	1	1
$\perp$	1	$\perp$	$\perp, 1.$

Now, what *inferences* can be justified by the *only* means of the informational meaning of the logical operators as specified by the informational 3-valued tables? Let $\mathcal{L}$ be a Boolean language with the four standard logical operators and $\mathsf{Form}(\mathcal{L})$ be the set of all $\mathcal{L}$-formulae.

Definition 2.1. *A* 3ND-valuation *is a mapping* $V : \mathsf{Form}(\mathcal{L}) \rightarrow \{0, 1, \perp\}$*, satisfying the following conditions for all* $A, B \in \mathsf{Form}(\mathcal{L})$*:*

1. $V(\neg A) = \hat{f}_\neg(V(A))$
2. $V(A \circ B) \in \hat{f}_\circ(V(A), V(B))$

where (i) $\circ$ *is* $\wedge$*,* $\vee$ *or* $\rightarrow$*, (ii)* $\hat{f}_\neg$ *is the deterministic truth-function defined by the informational 3-valued table for* $\neg$*, and (iii)* $\hat{f}_\circ$ *is the non-deterministic truth-function defined by the informational 3-valued table for* $\circ$*.*

This approach can be extended to arbitrary Boolean operators. A general method can be obtained from [21] by translating the modular semantics for arbitrary Boolean operators in terms of non-deterministic truth-functions (Section 2.9) that satisfy Conditions 2 and 3. A 3ND-valuation can be seen as describing a minimal *information state* that is closed under the implicit information that depends only on the informational meaning of the logical operators. This is information that we *actually possess* and with which we can operate, in the precise sense that we have (as will be shown in the sequel) a *natural* and *feasible* procedure to decide, for every formula A, whether the information that A is true, or the information that A is false, or neither of them actually belongs to our information state.[10]

[9] This is just convenient jargon for a function $V^2 \rightarrow 2^V \setminus \{\emptyset\}$, with V the set of truth-values.

[10] Assuming that $P \neq NP$, this is clearly not the case of the implicit information that stems from their classical meaning.

It may be observed that: (i) agents may not be *aware* even of easy consequences of their assumptions and there is still a difference between *implicit* and *explicit* information,[11] (ii) even implicit information that can be feasibly extracted from the explicit one requires the consumption of resources. Both observations raise interesting questions that we do not address here. We focus on the distinction between *two kinds of implicit information*: the first type is the one that can be feasibly extracted by using only information that we actually possess (*actual information*); the second is the one that essentially requires the simulation of potential information that we do not actually possess (*virtual information*). It turns out that the latter can also be feasibly extracted whenever the nested use of virtual informaiton is limited.

In what follows we shall make use of *signed formulae* (*S-formulae* for short), namely expressions of the form $T\,A$ or $F\,A$ with the intended meaning of "A is informationally true" or "we actually possess the information that A is true" and "A is informationally false" or "we actually possess the information that A is false".[12] Using signed formulae allows us to express a 3ND-valuation V as a set of S-formulae, namely the set $\{T\,A \mid V(A) = 1\} \cup \{F\,A \mid V(A) = 0\}$. We shall use "$\varphi, \psi, \theta, \ldots$", as variables ranging over S-formulae and continue using "$A, B, C, \ldots$" as variables ranging over usual unsigned formulae. We shall also use "$X, Y, Z, \ldots$", as variables ranging over sets of S-formulae and continue using "$\Gamma, \Delta, \Lambda, \ldots$", as variables ranging over sets of unsigned formulae. The *unsigned part* of an S-formula is the unsigned formula that results from it by removing the sign T or F. Given an S-formula φ, we denote by φ^u the unsigned part of φ and by X^u the set $\{\varphi^u \mid \varphi \in X\}$.

Let us say that a 3ND-valuation V *satisfies* an S-formula $T\,A$ if $V(A) = 1$ and an S-formula $F\,A$ if $V(A) = 0$.

Definitions 2.2. *For every set X of S-formulae and every S-formula φ, we say that:*

- *φ is a* 0-depth consequence *of X if V satisfies φ for every 3ND-valuation V such that V satisfies all the S-formulae in X.*
- *X is* 0-depth inconsistent *if there is no 3ND-valuation V such that V satisfies all the S-formulae in X.*

In the sequel, we shall use the symbol "$\vDash_0$" for the 0-depth consequence relation and write "$X \vDash_0 \varphi$" for "φ is a 0-depth consequence of X". We shall

[11] This issue is interestingly related to the vast literature on *awareness* (see, for example, [42, 30, 41, 49]).

[12] In fact, in this context, the signs "T" and "F" act as propositional attitudes and, in a multiagent setting, can be indexed by symbols "$x, y, z, \ldots$" standing for different agents. So $T_x\,A$ and $F_x\,A$ mean that A is informationally true, respectively false, for agent x. In this paper we shall omit indexes since we will not be dealing with multiagent systems. From this point of view one could see our signs as a sort of epistemic modalities and our consequence relations as a sort of first-degree epistemic logics in which the modalities cannot be iterated or used within a sentence.

also write $X \vDash_0$ to mean that X is 0-depth inconsistent. The notions of 0-depth consequence and 0-depth inconsistency can be extended to unsigned formulae as follows (writing $T\,\Gamma$ for $\{T\,A \mid A \in \Gamma\}$):

Definitions 2.3. *For every set Γ of unsigned formulae and every unsigned formula A, we say that:*

- *A is a 0-depth consequence of a set Γ if $T\,\Gamma \vDash_0 T\,A$*
- *Γ is 0-depth inconsistent if $T\,\Gamma$ is 0-depth inconsistent.*

We shall abuse of the same relation symbol "$\vDash_0$" to denote 0-depth consequence and inconsistency for both signed and unsigned formulae. In [21] (Proposition 2.49) it is shown that 0-depth consequence and 0-depth inconsistency cannot be characterized by any finite deterministic matrix. So, the logic $\vDash_0$ is not a finite many-valued logic in the standard sense.

The 0-depth consequence relation $\vDash_0$ is a subsystem of classical propositional logic obtained by replacing the notion of "possible world" with our weaker notion of information state (described by a 3ND-valuation). It is not difficult to show that the relation $\vDash_0$ is a *Tarskian consequence relation*, that is, it satisfies reflexivity, monotonicity and cut. It is also *structural*, in that it satisfies substitution invariance. Like Kleene's 3-valued logic [40, §64] and Belnap's 4-valued logic [6, 7], this consequence relation has *no tautologies*.[13]

An important consequence of its informational characterization is that the 0-depth logic $\vDash_0$ is *tractable*, just as we should expect given that it intends to be the logic of "actual information", namely the information that is practically available to an agent (as opposed to the potential information that is available to her only in principle).[14] However, this is far from being obvious if we focus on the 3ND-table presentation, which seems to suggest an exponential blow up as in the classical case. Indeed, the tractability of the 0-depth logic will become apparent in Section 6, when we shall provide a natural proof-theoretical characterization of this logic that is quite close to deductive practice.

Observe that, according to our definitions, $\vDash_0$ is *explosive* just like classical logic: when X is 0-depth inconsistent, $X \vDash_0 \varphi$ for every φ, since there is no 3ND-valuation V that satisfies all the formulae in X. However, 0-depth inconsistency is stricter than classical inconsistency — a set X of S-formulae may well be 0-depth consistent but classically inconsistent — and, more importantly, can be feasibly detected (see Proposition 6.1 below and the following comment).

[13] This is not so surprising if one thinks that a tautology is usually described as "a consequence of the empty set of premises". There is no way of extracting information from the empty information state without simulating virtual extensions of it. Accordingly, tautologies make their appearance only at depths $k > 0$, when the use of virtual information is allowed, and the set of provable tautologies increases with k.

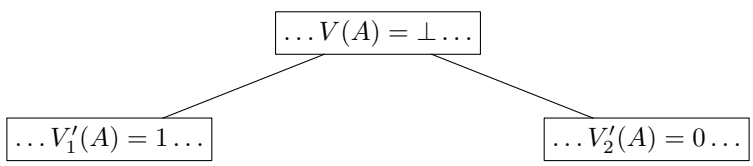
[14] We stress that tractability here is, so to speak, a "side effect" of an informational interpretation of the logical operators that makes no direct reference to questions of computational complexity. On this point see also footnote 26 and related text.

3. Virtual information and depth-bounded consequence

What about the classical inferences that are not valid in the 0-depth logic? For example, consider the classically valid inference:

$$\frac{A \vee B \quad \neg A \vee B}{B}$$

This inference cannot be justified by the 3ND-tables. A counterexample is any 3ND-valuation V such that $V(A) = V(B) = \bot$ and $V(A \vee B) = V(\neg A \vee B) = 1$. In order to validate the above inference, we need to restrict our attention to the *refinements* of V in which the value of A is defined, namely the 3ND-valuations V' such that for all B, $V(B) \preceq V'(B)$ and $V'(A) \neq \bot$:



It is easy to check, using the 3ND-tables for $\vee$ and $\neg$, that $V'(B) = 1$ for every such refinement of V. The information concerning A in either of these refinements is not even implicitly contained in the actual information state expressed by V. This is what we call *virtual information*. So, the 0-depth logic is simply the logic of deductive reasoning with *no virtual information*.

The notion of k-depth consequence depends not only on the depth at which the use of virtual information is allowed, but also on the subset of $\mathsf{Form}(\mathcal{L})$ on which the introduction of virtual information is allowed. In [21] this subset was called the *virtual space* and, in the context of this paper, can be simply defined as a function f of the set $\Gamma \cup \{A\}$ consisting of the premises Γ and of the conclusion A of the the given inference.[15]

In the sequel we shall denote by "$\mathtt{sub}$" the function that maps any given set Δ of formulae to the set of all its subformulae, and by "$\mathtt{at}$" the function that maps any given Δ to the set of its atomic subformulae. Let $\mathcal{F}$ be the set of all operations f on the finite subsets of $\mathsf{Form}(\mathcal{L})$ such that: (i) for all Δ, $\mathtt{at}(\Delta) \subseteq f(\Delta)$, (ii) $f(\Delta)$ is closed under subformulae, that is, $\mathtt{sub}(f(\Delta)) = f(\Delta)$, (iii) $|f(\Delta)| \leq p(|\Delta|)$ for some fixed polynomial p, where we denote by $|\Lambda|$ the number of occurrences of symbols in Λ (the *size* of Λ).[16] Distinguished examples of operations in $\mathcal{F}$ are $\mathtt{sub}$ and $\mathtt{at}$. However, in general, $f(\Delta)$ may contain also formulae that are not in $\mathtt{sub}(\Delta)$. For example, the operation f that maps Δ to the set of all formulae of bounded logical complexity that can be built out of $\mathtt{sub}(\Delta)$ or of $\mathtt{at}(\Delta)$ is also in $\mathcal{F}$. The operations in $\mathcal{F}$ are partially

[15] In [21] the virtual space was defined as a function of the search space that was in turn a function of $\Gamma \cup \{A\}$ (Section 3.2).

[16] This third requirement is essential in order to define a hierarchy of *tractable* approximations to Boolean logic.

ordered by the relation $\trianglelefteq$ such that $f_1 \trianglelefteq f_2$ if and only if, for every finite Δ, $f_1(\Delta) \subseteq f_2(\Delta)$.

Definition 3.1. *For all X, φ, and for all $f \in \mathcal{F}$,*

1. $X \vDash_0^f \varphi$ *if and only if* $X \vDash_0 \varphi$*;*
2. $X \vDash_{k+1}^f \varphi$ *if and only if* $X \cup \{T\,A\} \vDash_k^f \varphi$ *and* $X \cup \{F\,A\} \vDash_k^f \varphi$ *for some* $A \in f(X^u \cup \{\varphi^u\})$.

Notice that the above definition covers also the case of k-depth inconsistency by assuming $X \vDash_k^f$ as equivalent to $X \vDash_k^f \varphi$ for all φ. So:

1. $X \vDash_0^f$ if and only if $X \vDash_0$;
2. $X \vDash_{k+1}^f$ if and only if $X \cup \{T\,A\} \vDash_k^f$ and $X \cup \{F\,A\} \vDash_k^f$ for some $A \in f(X^u \cup \{\varphi^u\})$.

When $X \vDash_k^f \varphi$ ($X \vDash_k^f$) we say that φ *is a k-depth consequence of X (X is k-depth inconsistent) over the f-bounded virtual space.* Observe that, since $\vDash_0$ is monotonic,

$$\vDash_j^f \subseteq \vDash_k^f \text{ whenever } j \leq k. \tag{4}$$

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. Observe also that:

$$\vDash_k^{f_1} \subseteq \vDash_k^{f_2} \text{ whenever } f_1 \trianglelefteq f_2. \tag{5}$$

Then, it is not difficult to show that:

Proposition 3.2. *For every f, the relation $\vDash_\infty^f = \bigcup_{k \in \mathbb{N}} \vDash_k^f$ is the consequence relation of classical propositional logic.*

Proof. Suppose that Γ classically implies A and let $p_1, \ldots, p_k$ the atomic formulae occurring in $\Gamma \cup \{A\}$. Let V be an arbitrary 3ND-valuation such that (i) $V(A) = 1$ for all $A \in \Gamma$ and (ii) $V(p_i) \neq \bot$ for all $i = 1, \ldots, k$. Since the 3ND-tables agree with the classical truth-tables whenever the rows consist all of defined values, and Γ classically implies φ, it follows that $V(A) = 1$. This implies, by definition of $\vDash_k^{\mathtt{at}}$, that $\Gamma \vDash_k^{\mathtt{at}} A$. Since, by definition of $\mathcal{F}$, $\mathtt{at} \trianglelefteq f$ for every $f \in \mathcal{F}$, it follows from (5) that $\Gamma \vDash_k^f A$ for every f. $\square$

While the 0-depth logic is Tarskian and structural, the k-depth consequence relations are not transitive[17] and may not be structural. Unbounded transitivity is replaced by its depth-bounded version:

$$\frac{X \vDash_j^f \varphi \quad X, \varphi \vDash_k^f \psi}{X \vDash_{j+k}^f \psi}$$

[17] In the terminology of [21] these are "weak depth-bounded consequence relations", see Section 3.3.

Structurality depends on the function f that defines the virtual space. For example $\vDash_k^{at}$ is *not* structural. While $\emptyset \vDash_1^{\mathtt{at}} p \vee \neg p$, the minimum depth k at which $\emptyset \vDash_1^{\mathtt{at}} \sigma(p \vee \neg p)$ depends on the substitution σ. On the other hand $\vDash_k^{\mathtt{sub}}$ *is* structural. In general, structurality can be imposed by restricting the operations in $\mathcal{F}$ to those such that, for every σ, Δ, $\sigma f(\Delta) \subseteq f(\sigma\Delta)$, where by $\sigma\Lambda$ we mean the result of applying σ to every formula in Λ. This is not satisfied when $f = \mathtt{at}$. However, it is satisfied when $f(\Delta) = \mathtt{sub}(\Delta)$ or $f(\Delta)$ is the set of all formulae of given bounded complexity that can be built out of $\mathtt{sub}(\Delta)$. As will be shown in the next section, each $\vDash_k^f$ inherits the tractability of $\vDash_0$ although the complexity of the natural decision procedure grows with k (and with the degree of the polynomial p that bounds the size of the virtual space defined by f).

4. Classical Intelim Deduction

A natural proof-theoretical characterization of the 0-depth logic $\vDash_0$ is obtained by means of a set of introduction and elimination rules (*intelim* rules) for the logical operators that are are displayed in Tables 2 and 3. In view of the informational interpretation of the signs T and F (see p. 8 above), as expressing the informational truth and the informational falsity of the sentence to which they are prefixed, the intelim rules are presented in terms of S-formulae, to highlight their correspondence with the informational semantics of the previous sections. However, a version for unsigned formulae is simply obtained by replacing each S-formula $T\,A$ with A and each S-formula $F\,A$ with $\neg A$.

Their soundness can be immediately verified by inspection of the 3ND-tables. For example, if an agent x actually possesses the information that $A \vee B$ is true (the value of $A \vee B$ is 1) and x actually possesses the information that A is false, (the value of $\neg A$ is 0), then x actually possesses also the information that B is true, since the other possible two values are ruled out by the table for $\vee$. It turns out that the intelim rules are also complete for the 0-depth logic, as will be shown later on.

Our intelim rules are different from the standard intelim rules of Gentzen-style natural deduction and are better suited to represent arguments in classical logic. In this respect, 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. Accordingly, for each logical operator, we have intelim rules for the truth of a sentence containing it as main operator and intelim rules for the falsity of such a sentence.[18] In these rules the sentence containing the logical operator that is to be eliminated is called *major premise* and the other is called *minor premise*.

[18] Observe that the two-premise elimination rules for true disjunctions and false conjunctions correspond to time-honoured principles of inference: *modus ponens*, *modus tollens*, *disjunctive syllogism* and its dual. The less natural rules, from the point of view of ordinary usage, namely the introduction rules for true conditionals and the elimination rules for false conditionals, are however faithful to the classical "truth-table" meaning of this operator. On the other hand Gentzen's rule for introducing the conditional is faithful to its intuitionistic meaning.

$$\frac{F\ A}{T\ \neg A}\ T\neg\text{-}\mathcal{I} \qquad \frac{T\ A}{F\ \neg A}\ F\neg\text{-}\mathcal{I}$$

$$\frac{T\ A}{T\ A \vee B}\ T\vee\text{-}\mathcal{I}1 \qquad \frac{T\ B}{T\ A \vee B}\ T\vee\text{-}\mathcal{I}2 \qquad \frac{\begin{array}{c}F\ A\\ F\ B\end{array}}{F\ A \vee B}\ F\vee\text{-}\mathcal{I}$$

$$\frac{F\ A}{F\ A \wedge B}\ F\wedge\text{-}\mathcal{I}1 \qquad \frac{F\ B}{F\ A \wedge B}\ F\wedge\text{-}\mathcal{I}2 \qquad \frac{\begin{array}{c}T\ A\\ T\ B\end{array}}{T\ A \wedge B}\ T\wedge\text{-}\mathcal{I}$$

$$\frac{F\ A}{T\ A \to B}\ T\to\text{-}\mathcal{I}1 \qquad \frac{T\ B}{T\ A \to B}\ T\to\text{-}\mathcal{I}2 \qquad \frac{\begin{array}{c}T\ A\\ F\ B\end{array}}{F\ A \to B}\ F\to\text{-}\mathcal{I}$$

Table 2: Introduction rules for the standard Boolean operators.

$$\frac{T\ \neg A}{F\ A}\ T\neg\text{-}\mathcal{E} \qquad \frac{F\ \neg A}{T\ A}\ F\neg\text{-}\mathcal{E}$$

$$\frac{\begin{array}{c}T\ A \vee B\\ F\ A\end{array}}{T\ B}\ T\vee\text{-}\mathcal{E}1 \qquad \frac{\begin{array}{c}T\ A \vee B\\ F\ B\end{array}}{T\ A}\ T\vee\text{-}\mathcal{E}2 \qquad \frac{F\ A \vee B}{F\ A}\ F\vee\text{-}\mathcal{E}1 \qquad \frac{F\ A \vee B}{F\ B}\ F\vee\text{-}\mathcal{E}2$$

$$\frac{\begin{array}{c}F\ A \wedge B\\ T\ A\end{array}}{F\ B}\ F\wedge\text{-}\mathcal{E}1 \qquad \frac{\begin{array}{c}F\ A \wedge B\\ T\ B\end{array}}{F\ A}\ F\wedge\text{-}\mathcal{E}2 \qquad \frac{T\ A \wedge B}{T\ A}\ T\wedge\text{-}\mathcal{E}1 \qquad \frac{T\ A \wedge B}{T\ B}\ T\wedge\text{-}\mathcal{E}2$$

$$\frac{\begin{array}{c}T\ A \to B\\ T\ A\end{array}}{T\ B}\ T\to\text{-}\mathcal{E}1 \qquad \frac{\begin{array}{c}T\ A \to B\\ F\ B\end{array}}{F\ A}\ T\to\text{-}\mathcal{E}2 \qquad \frac{F\ A \to B}{T\ A}\ F\to\text{-}\mathcal{E}1 \qquad \frac{F\ A \to B}{F\ B}\ F\to\text{-}\mathcal{E}2$$

Table 3: Elimination rules for the four standard Boolean operators

1	$T\,(p \vee q) \to \neg r$	Assumption
2	$T\,p$	Assumption
3	$T\,(p \wedge t) \to r$	Assumption
4	$T\,p \vee q$	$T\vee$-$\mathcal{I}1$ (2)
5	$T\,\neg r$	$T \to$-$\mathcal{E}1$ (1,4)
6	$F\,r$	$T\neg$-$\mathcal{E}$ (5)
7	$F\,(p \wedge t)$	$T \to$-$\mathcal{E}2$ (3,6)
8	$F\,t$	$F\wedge$-$\mathcal{E}1$ (7,2).

1	$(p \vee q) \to \neg r$	Assumption
2	p	Assumption
3	$(p \wedge t) \to r$	Assumption
4	$p \vee q$	$T\vee$-$\mathcal{I}1$ (2)
5	$\neg r$	$T \to$-$\mathcal{E}1$ (1,4)
6	$\neg(p \wedge t)$	$T \to$-$\mathcal{E}2$ (3,5)
7	$\neg t$	$F\wedge$-$\mathcal{E}1$ (6,2).

Figure 1: On the left, an intelim sequence which proves the S-formula on line 8 from the assumptions, using the rules for signed formulae. On the right, the corresponding sequence using the rules for unsigned formulae.

The intelim rules generate *intelim sequences*, i.e., 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 formulae. In Figure 1 we show simple examples of intelim sequences using, respectively, the intelim rules for signed formulae and their version for unsigned formulae. The intelim rules are *not* complete for Boolean logic, but only for the 0-depth consequence relation $\vDash_0$ (this will be shown later on in Proposition 4.7). Completeness for full Boolean logic is obtained by adding only the following *branching* rule:[19]

$$\begin{array}{ccc} & & \text{PB} \\ / & \backslash & \\ T\,A & & F\,A \end{array}$$

With the addition of PB to the stock of rules, proofs and refutations are represented, as in semantic tableaux, by downward-growing *intelim trees*.

Each application of PB invites us to consider virtual information about the truth or falsity of the formula A (the *PB-formula*) and allows us to append both $T\,A$ and $F\,A$ as sibling nodes at the end of any branch of the tree, generating two new branches. The S-formulae $T\,A$ and $F\,A$ are called *virtual assumptions*. Notice that PB is, in essence, a classical cut rule which is *not eliminable*, but whose use (as will be shown in the sequel) can be restricted so as to satisfy the subformula property. The main conceptual advantage of this proof-theoretical characterization of classical logic, from our informational viewpoint, consists in the fact that it clearly separates the rules that fix the meaning of the logical operators in terms of the information that we actually possess (the intelim rules) from the single structural rule that introduces virtual information (the PB rule).[20] Intuitively, the more virtual information needs to be invoked via PB, the more difficult the deductive process is both from the computational and the

[19] "PB" stands for "Principle of Bivalence".

[20] By contrast, in Gentzen-style systems some of the intelim rules (the "discharge rules" of natural deduction and their counterparts in the sequent calculus) make essential use of virtual information. Since in Gentzen-style proof systems cut is eliminable, no approximation hierarchy can be produced by controlling the application of the cut rule.

cognitive viewpoint.[21] The method of intelim trees bears some resemblance with Smullyan-style semantic tableaux [24]. However: (i) like in Natural Deduction there are introduction as well as elimination rules, and the method can be used as a direct proof method as well as a refutation method, (ii) the tableau branching rules are replaced by two-premise rules, so that all the operational rules have a linear format, (iii) there is only one branching rule corresponding to the Principle of Bivalence. A variant which brings out the analogy with Gentzen-style natural deduction is described in the Appendix.

Definition 4.1. *An* intelim tree *for X is a finite tree $\mathcal{T}$ of S-formulae such that, for every S-formula φ occurring in $\mathcal{T}$, either (i) $\varphi \in X$, or (ii) φ results from an application of an intelim rule to preceding S-formulae in the same branch, or (iii) φ is a virtual assumption introduced by an application of the branching rule PB.*

We say that a branch of an intelim tree is *closed* if it contains both $T\,A$ and $F\,A$ for some formula A, otherwise it is *open*.

Definitions 4.2. *For all X, φ,*

1. *An* intelim proof *of φ from X is an intelim tree $\mathcal{T}$ for X such that φ occurs at the end of all open branches of $\mathcal{T}$;*
2. *A refutation of X is an intelim tree $\mathcal{T}$ for X such that every branch of $\mathcal{T}$ is closed.*

Notice that, according to the above definition, every refutation of X is, at the same time, a proof of φ from X, for every S-formula φ (since there are no open branches and the condition that φ occurs at the end of all open branches is vacuously satisfied).

Definition 4.3. *We say that an intelim proof of φ from X (an intelim refutation of X) has the* subformula property *(SFP) if, for every S-formula ψ occurring in it, ψ^u is a subformula of θ^u for some θ in $X \cup \{\varphi\}$ (in X).*

In the next section we shall show that every intelim proof of φ from X can be transformed into an intelim proof of φ from X with the SFP. The SFP 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 only inference steps involving formulae that are "contained" in the assumptions (or also in the conclusion in the case

[21] PB is a form of non-constructive dilemma. In a recent interesting paper [39], Klassen and colleagues propose a similar view of classical case analysis as the source of the "effort" required by a deductive task. This idea was already present in some early contributions on approximation methods [13, p. 286]; see also the notion of "intricacy" in [26]. In our view, the primary source of intractability is the use of virtual information, a more general phenomenon than reasoning by cases. Indeed, the pure implication fragment of intuitionistic logic, which is characterized simply by the NJ intelim rules for $\rightarrow$, is P-SPACE complete [50], but does not appear to involve any case analysis. It does, however, make use of virtual information (in the introduction rule for $\rightarrow$).

of proofs). So, no special ingenuity is required to construct such an analytic argument and its search is amenable to algorithmic treatment. In particular, in our classical intelim system, the SFP guarantees that we can impose a bound on the applications of PB, which could in principle be applied to arbitrary formulae, with no loss of deductive power. Similarly, we can impose a bound on the sensible applications of introduction rules, which could in principle be applied *ad infinitum* leading to ever more complex formulae. (On this point see Proposition 5.10 below.)

When moving from intelim trees for S-formulae to trees for standard unsigned formulae, the subformula property is weakened as follows:

Definition 4.4. *We say that an intelim proof $\mathcal{T}$ of A from Γ has the* weak subformula property *(WSFP) if every formula occurring in $\mathcal{T}$ is a weak subformula*[22] *of some formula in $\Gamma \cup \{A\}$.*

Definitions 4.5. *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 refutation of X*) if $\mathcal{T}$ is an intelim proof of φ from X (a refutation of X) and $\mathcal{T}$ is of depth k.*

Observe also that a 0-depth intelim tree is nothing but an *intelim sequence*. An example of an intelim proof of depth 2 with the SFP is given in Figure 2. This is an intelim proof of $T\,u$ from the premises marked with a "$*$". The reader can check that each S-formula that is not a premise either is obtained from previous S-formulae on the same branch by an application of one of the intelim rules in Tables 2 and 3, or is one of the virtual assumptions introduced by the branching rule PB. All the open branches end with the S-formula $T\,u$. The rightmost branch is closed since it contains both $T\,r$ and $F\,r$. Each open branch is a 0-depth intelim proof of $T\,u$ from the union of the initial premises plus the virtual assumptions introduced by the rule PB on that branch. In Fig 3 we show three examples of intelim trees using unsigned formulae that enjoy the WSFP. Again, the initial assumptions are the formulae marked with a "$*$".

Let us first focus on 0-depth intelim proofs and refutations.

Definition 4.6. *For all X, φ, we say that φ is* 0-depth deducible from X*, and write $X \vdash_0 \varphi$, if there is a 0-depth proof of φ from X. We also say that X is* 0-depth refutable*, and write $X \vdash_0$, if there is a 0-depth refutation of X.*

Proposition 4.7. *For every set X of S-formulae and every S-formula φ,*

$$X \vDash_0 \varphi \text{ if and only if } X \vdash_0 \varphi.$$

Proof. An indirect proof is given in [21] where the 0-depth consequence relation $\vDash_0$ is characterized in terms of another semantics (called "modular semantics")

[22] Recall that B is a weak subformula of A if B is a subformula of A or the negation of a subformula of A.

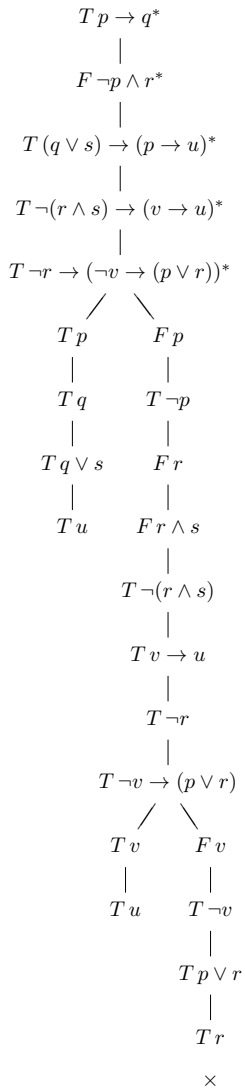


Figure 2: An intelim proof of depth 2 using S-formulae. Each branch is an intelim sequence for the set of S-formulae containing the initial assumptions (marked with $*$) and the virtual assumptions introduced by the applications of PB.

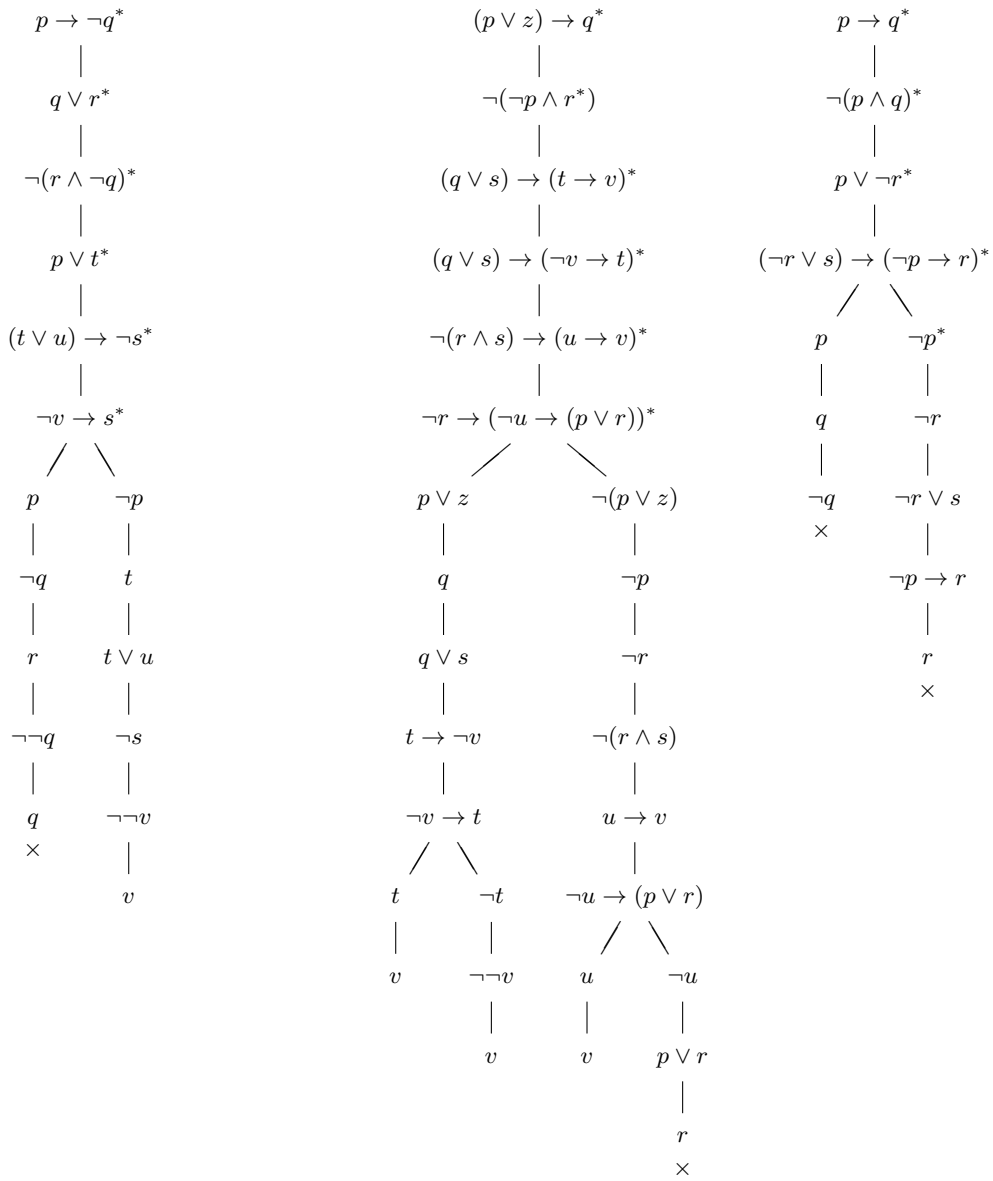


Figure 3: Intelim trees using unsigned formulae.

that is shown to be equivalent to the informational 3-valued semantics. Here we provide a direct adequacy proof.

The reader can check that the intelim rules are all sound with respect to the informational 3ND-tables. 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 $X \vdash_0 \varphi$ against the hypothesis. Now, consider the set $X^* = \{\psi \mid X \vdash_0 \psi\}$. Since X is not 0-depth refutable, for no formula A, $T\,A$ and $F\,A$ are both in X^*. Then, it is easy to verifty that the function V defined as follows:

$$V(A) = \begin{cases} 1 & \text{if } T\,A \in X^* \\ 0 & \text{if } F\,A \in X^* \\ \bot & \text{otherwise} \end{cases}$$

is a 3ND-valuation, i.e. it agrees with the 3-valued informational tables. Here we just outline a typical case. Suppose $V(A) = V(B) = \bot$. Then $F\,A \vee B \notin X^*$. Otherwise, if $F\,A \vee B \in X^*$, then by definition of X^* and by the rules $F\vee$-$\mathcal{E}$, $F\,A$ and $F\,B$ should also be in X^*; therefore, by definition of V, $V(A) = V(B) = 0$ against our assumption. Hence $V(A \vee B) \neq 0$. Moreover, $T\,A \vee B$, may or may not belong to X^*, and so $V(A \vee B) = 1$ or $V(A \vee B) = \bot$. Finally, observe that: (i) $\psi \in X^*$ for all $\psi \in X$ and so, by definition of V, V satisfies all $\psi \in X$; (ii) by the hypothesis that $\Gamma \nvdash_0 \varphi$, $\varphi \notin X^*$ and so V does not satisfy φ. Hence $X \nvDash_0 \varphi$. □

Corollary 4.8. *For every set X of S-formulae,*

$$X \vDash_0 \text{ if and only if } X \vdash_0 .$$

The following definition mimics Definition 3.1:

Definition 4.9. *For all X, φ and all $f \in \mathcal{F}$,*

1. *$X \vdash_0^f \varphi$ if and only if $X \vdash_0 \varphi$;*
2. *$X \vdash_{k+1}^f \varphi$ if and only if $X \cup \{T\,A\} \vdash_k^f \varphi$ and $X \cup \{F\,A\} \vdash_k^f \varphi$ for some $A \in f(X^u \cup \{\varphi^u\})$;*

As Definition 3.1, the above definition covers the case of k-depth refutability. When $X \vdash_k^f \varphi$ ($X \vdash_k^f$) we say that φ is *deducible at depth k from X (X is refutable at depth k) over the f-bounded virtual space*. It follows immediately from Definitions 4.9 and 4.5 that:

Proposition 4.10. *For all X, φ and all $f \in \mathcal{F}$, $X \vdash_k^f \varphi$ ($X \vdash_k^f$) if and only if there is a k-depth intelim proof of φ from X (a k-depth intelim refutation of X) such that all its PB-formulae are in $f(X^u \cup \{\varphi^u\})$.*

Given Proposition 4.7 and the close correspondence between Definitions 3.1 and 4.9, it is far from surprising that:

Proposition 4.11. *For all X, φ, $X \vDash_k^f \varphi$ if and only if $X \vdash_k^f \varphi$.*

5. Normal intelim proofs

Consider the following intelim sequences:

1	$T\,p \to \neg q$	Assumption
2	$T\,(p \to \neg q) \to p$	Assumption
3	$T\,p \to r$	Assumption
4	$T\,p$	Assumption
5	$T\,\neg q$	from 1,4
6	$F\,q$	from 5
7	$T\,p \vee q$	from 4
8	$T\,p$	from 4,6
9	$T\,r$	from 3,7

1	$T\,p$	Assumption
2	$T\,\neg p$	Assumption
3	$F\,p$	from 2
4	$T\,p \vee q$	from 1
5	$T\,q$	from 4, 3

The first one is an intelim proof of $T\,r$ from $\{T\,p \to \neg q, T\,(p \to \neg q) \to p, T\,p \to r\}$ and the second one is the so-called "Lewis"' proof of (an arbitrary) $T\,q$ from $\{T\,p, T\,\neg p\}$, which is often used to show the explosivity of classical logic. Observe that both proofs are *redundant*.

In the lefthand proof the S-formula $T\,p \vee q$ is first introduced (from premise $T\,p$) and then eliminated (using the minor premise $F\,q$) to re-obtain the S-formula $T\,p$ which was *already* contained in the sequence, that is, this proof contains circular reasoning. In the righthand proof, the S-formula $T\,p \vee q$ is first introduced (from premise $T\,p$) and then eliminated (using $F\,p$ as minor premise); however, the sequence was already closed before the $T\vee$-introduction and so, by Definition 4.2.1, the closed sequence $T\,p, T\,\neg p, F\,p$ was *already* a proof of $T\,q$ from $T\,p$ and $T\,\neg p$.

The same kind of redundancy is observed whenever a formula is, at the same time, the conclusion of an introduction and the major premise of an elimination.

Definition 5.1. *We say that an occurrence of an S-formula φ in an intelim tree $\mathcal{T}$ is a* detour *if φ is both the conclusion of an introduction and the major premise of an elimination.*

Let $\overline{\varphi}$ denote the *conjugate* of the S-formula φ, namely the S-formula $T\,A$ if φ is equal to $F\,A$ and $F\,A$ if φ is equal to $T\,A$.

Definition 5.2. *An occurrence of an S-formula φ is* idle *in an intelim tree $\mathcal{T}$ if (i) it is not the terminal node of its branch, (ii) φ is not used in $\mathcal{T}$ as premise of some application of an intelim rule, and (iii) it is not the conjugate of some S-formula occurring in the same branch.*

Definition 5.3. *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 $\overline{\varphi}$ for some formula φ.*

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

Definition 5.4. *Let $\mathcal{T}$ be an intelim proof of φ from Γ (an intelim refutation of Γ). We say that $\mathcal{T}$ is* non-redundant *if it satisfies the following conditions:*

1. *$\mathcal{T}$ contains no idle occurrences of formulae;*
2. *no branch of $\mathcal{T}$ contains more than one occurrence of the same formula;*
3. *no branch of $\mathcal{T}$ properly includes a closed path.*

Our argument above shows that whenever an intelim proof or refutation $\mathcal{T}$ contains a detour, then either the second or the third non-redundancy condition is violated. Thus:

Lemma 5.5. *If an intelim proof or refutation $\mathcal{T}$ is non-redundant, then it contains no detours.*

Proof. Suppose $\mathcal{T}$ contains a detour, namely a formula φ that is at the same time the conclusion of an introduction and the major premise of an elimination. By inspection of the rules, either the conclusion of the elimination is equal to one of the premises of the introduction, or the minor premise of the elimination is the complement of one of the premises of the introduction and so the branch was already closed before the elimination. In either case $\mathcal{T}$ is redundant. □

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

1. checking if there are closed paths and removing whatever follows;
2. removing any repetition of (S-)formulae in the same branch;
3. checking if there are idle occurrences of S-formulae, and
4. for each idle occurrence of an S-formula φ:
   (a) if φ is the conclusion of an application of an intelim rule, just remove φ from $\mathcal{T}$;
   (b) if φ is a virtual assumption introduced by an application of PB, remove both φ and the whole subtree generated by its conjugate S-formula $\overline{\varphi}$ introduced in the same application of 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.

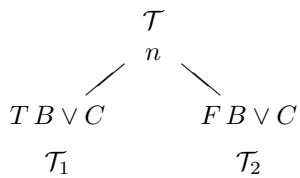
Given an intelim proof $\mathcal{T}$ of φ from X (an intelim refutation of X), and any operation $f \in \mathcal{F}$ (see Section 3 above), we say that an application of PB in $\mathcal{T}$ is *f-analytic* if its PB-formula is in $f(X^u \cup \{\varphi^u\})$ ($f(X^u)$), namely in the virtual space defined by the operation f (recall that this is, by definition, closed under subformulae and polynomially bounded). When $f = \texttt{sub}$, that is the virtual space consists exactly of the subformulae of $X^u \cup \{\varphi^u\}$ (of X^u), we just say that the application of PB is *analytic*. Then, it can be shown that:

Lemma 5.6. *Given any $f \in \mathcal{F}$, every k-depth intelim proof $\mathcal{T}$ of φ from X (k-depth intelim refutation of X) can be transformed into a $k+j$-depth intelim proof $\mathcal{T}'$ of φ from X (intelim refutation $\mathcal{T}'$ of X), for some $j \geq 0$, such that every application of PB in $\mathcal{T}$ is f-analytic.*

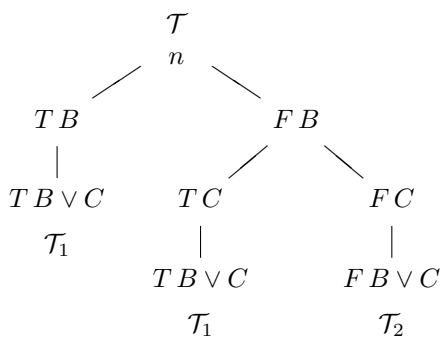
Proof. We use the notation $\begin{matrix}\mathcal{T}\\ n\end{matrix}$ to denote either an empty intelim tree or a non-empty intelim tree such that n is one of its terminal nodes. The proof is by lexicographic induction on $\langle \gamma(\mathcal{T}), \lambda(\mathcal{T})\rangle$, where $\gamma(\mathcal{T})$ is the maximum logical complexity[23] of a PB-formula in $\mathcal{T}$ that is not f-analytic and $\lambda(\mathcal{T})$ is the number of occurrences of such non-f-analytic PB-formulae of maximal complexity.

Let $\gamma(\mathcal{T}) = m > 0$ and let A be a PB-formula of logical complexity m. There are several cases depending on the logical form of A. We discuss only the case $A = B \vee C$, the other cases being similar.

If $A = B \vee C$, then $\mathcal{T}$ has the following form:



where $\mathcal{T}_1$ and $\mathcal{T}_2$ are intelim trees such that each of their open branches contains φ (or are both closed intelim trees in case $\mathcal{T}$ is a refutation of X). Let $\mathcal{T}'$ be the following intelim tree:



[23] The logical complexity of a formula is the number of occurrences of logical operators in it.

Clearly $\mathcal{T}'$ is a $k+1$-depth intelim proof of φ from X (a $k+1$-depth intelim refutation of X). Moreover, either $\gamma(\mathcal{T}') < \gamma(\mathcal{T})$, or $\gamma(\mathcal{T}') = \gamma(\mathcal{T})$ and $\lambda(\mathcal{T}') < \lambda(\mathcal{T})$. $\square$

In fact, the construction used in the proof of the above proposition shows that every intelim tree can be transformed into an equivalent one in which all the PB-formulae are *atomic*. So, in principle, we could reformulate the notion of intelim tree in such a way that PB is applied only to atomic formulae without loss of completeness. However: (i) each application of this construction increases the depth of the tree, so that it is convenient to use it only to the extent in which it is needed to remove applications of PB that are not f-analytic; (ii) if we insist that the applications of 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. On the other hand, if we require that the notion of intelim tree be restricted so as to permit only *analytic* applications of PB (that is, f-analytic applications with $f = \texttt{sub}$) , the property of being an intelim tree is indeed invariant under uniform substitutions.

Definition 5.7. *Given any $f \in \mathcal{F}$, we say that an intelim proof $\mathcal{T}$ of φ from X is* f-normal *if (i) $\mathcal{T}$ is non-redundant, (ii) every application or PB in $\mathcal{T}$ is f-analytic and (iii) neither of the two conjugate S-formulae in a closed branch is the conclusion of an introduction.*

When every application of PB is *analytic* (i.e., with $f = \texttt{sub}$) we just say that $\mathcal{T}$ is *normal*. We stress that f-normality is an important generalization of normality in the context of depth-bounded approximations in that in some cases the minimum depth of a normal proof is greater than the minimum depth of an f-normal proof. In general, whenever $f_1 \lhd f_2$ the minimum depth of an f_1-normal proof may be greater than that of an f_2-normal proof.

Proposition 5.8. *Given any $f \in \mathcal{F}$, every intelim proof $\mathcal{T}$ of φ from X (intelim refutation of X) can be transformed into an f-normal one.*

Proof. By Lemma 5.6, $\mathcal{T}$ can be transformed into a intelim proof $\mathcal{T}'$ of φ from X (intelim refutation of X) such that all the applications of PB are f-analytic. As explained above, $\mathcal{T}'$ can be transformed into a non-redundant intelim proof $\mathcal{T}''$ of φ from X (intelim refutation of X) with no size increase. Finally, suppose that ψ and $\overline{\psi}$ both occur in a branch of $\mathcal{T}''$. First, notice that they cannot be both conclusions of introductions, for in this case one can easily verify, by inspection of the introduction rules, that the branch would properly contain a closed path and so $\mathcal{T}''$ would be redundant. For example, suppose that $\psi = T\,A \wedge B$ and both $T\,A \wedge B$ and $F\,A \wedge B$ are conclusions of introductions. Then, both $T\,A$ and $T\,B$ and at least one of $F\,A$ and $F\,B$ already belong to the same branch, and so the branch would properly contain a closed path, against the hypothesis that $\mathcal{T}''$ is non-redundant. Suppose now that only one of them is the conclusion of an introduction. Then, just observe that this introduction can be retracted and replaced by an elimination without introducing a new detour. For example, suppose that $\psi = T\,A \vee B$ and that ψ is the conclusion of an introduction.

Then either $T\,A$ or $T\,B$ occurs above in the same branch. But either of $F\,A$ and $F\,B$ can be appended to the branch as conclusion of an application of $F\vee$-$\mathcal{E}$ to $\overline{\psi}$, so as to obtain a closed branch. Moreover, this move introduces no new detour because $\overline{\psi}$, by hypothesis, is not the conclusion of an introduction. The argument is the same for the other possible logical forms of ψ. $\square$

The following proposition states a generalization of the SFP for intelim proofs and refutations:[24]

Proposition 5.9 (Generalized SFP)**.** *For every $f \in \mathcal{F}$, if $\mathcal{T}$ is an f-normal proof of φ from X, or an f-normal refutation of X, then for every S-formula ψ occurring in $\mathcal{T}$,*

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

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

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

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

Proof. Let $\mathcal{T}$ be an f-normal intelim proof of φ from X (refutation of X) and suppose that there are S-formulae ω in $\mathcal{T}$ such that $\omega^u \notin f(X^u \cup \{\varphi^u\}) \cup \mathtt{sub}(X^u \cup \{\varphi^u\})$ ($\omega^u \notin f(X^u) \cup \mathtt{sub}(X^u)$). Let us call such formulae *spurious*. Let ψ be a spurious formula of maximal logical complexity.[25] Then ψ cannot result from the application of an elimination rule, otherwise $\mathcal{T}$ would contain a more complex spurious formula, namely the major premise of this elimination. Moreover, since $\mathcal{T}$ is f-normal, no spurious formula can occur in it as a virtual assumption introduced by an application of PB, since f-normal intelim trees contain only f-analytic applications of PB. Therefore ψ must be the conclusion of an introduction. Since $\mathcal{T}$ is non-redundant, it contains no idle occurrences of formulae, and so either (i) $\psi = \overline{\theta}$ for some θ occurring in the same branch or (ii) ψ is used as a premise of a rule application. But both alternatives are impossible. The first alternative violates condition (iii) in the definition of an f-normal proof (Definition 5.7). For the second alternative, first observe that ψ cannot be the minor premise of an elimination, otherwise there would be again a more complex spurious formula 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 5.5, $\mathcal{T}$ would be redundant, against the hypothesis that $\mathcal{T}$ is f-normal. $\square$

In the special case in which $f = \mathtt{sub}$ we obtain the usual SFP. Observe that, since 0-depth intelim trees have no virtual assumptions, every normal 0-depth proof or refutation has the SFP. The above proposition can be adapted to trees of unsigned formulae in the obvious way and, when dealing with such tress, the SFP is replaced, as before, by the WSFP.

[24] Notice that whenever $\Delta \subseteq f(\Delta)$, then also $\mathtt{sub}(\Delta) \subseteq f(\Delta)$.
[25] See footnote 23 above.

Let $\vdash_N^f$ be the unbounded deducibility relation defined as follows: $X \vdash_N^f \varphi$ ($X \vdash_N^f$) if there is an f-normal intelim proof of φ from X (an f-normal refutation of X). Proposition 5.8 guarantees that $\vdash_N^f$ is complete for classical propositional logic. This implies, among other things, that the application of the introduction rules can be *goal-oriented* in the sense clarified by the following:

Proposition 5.10. *Let $\mathcal{T}$ be an f-normal proof of φ from X (refutation of X) and let $\psi_1, \ldots, \psi_n$ be a maximal sequence of formulae occurring in a branch of $\mathcal{T}$ such that, for every $i = 2, \ldots, n$, ψ_i is the conclusion of an application of an introduction rule to previous S-formulae in the sequence. Then one of the following holds true:*

1. *ψ_n is the minor premise of an elimination*
2. *$\psi_n = \varphi$*

Proof. First, by Clause (iii) in Definition 5.7, ψ_n cannot be the conjugate of any other S-formula in the same branch. Since, by Clause (i), the intelim tree is non-redundant, ψ_n cannot be idle, and so it is either the terminal node of its branch, in which case the branch is open and $\psi_n = \varphi$, or the premise of an application of an intelim rule. Given that the sequence of introductions is maximal in the branch, ψ_n is not used as premise of an introduction. Moreover, by Proposition 5.5, non redundant proofs contain no detours, and so ψ_n is not used as major premise of an elimination. Thus either $\psi_n = \varphi$ of ψ is used as minor premise of an elimination. □

By the above propositions, the search for a proof or a refutation can be governed by a procedure that is informally described by the following four general rules for expanding an intelim tree:

1. stop expanding a branch whenever it is closed,
2. give priority to the elimination rules,
3. apply the introduction rules only to obtain either the conclusion of the proof, or a minor premise that is needed for an elimination;
4. apply the branching rule PB to an open branch only when instructions 1–3 fail.

The choice of the PB-formula in the last instruction depends on the operation f that defines the virtual space. When $f = \texttt{sub}$, one can always, without loss of completeness, choose as PB-formula some subformula of the assumptions or of the conclusion that does not already occur in the branch. This is the procedure that has been followed in the construction of the trees in Fig. 3. If we apply these rules mechanically, the resulting intelim proof would not contain detours but may still be redundant in that it may contain idle formulae. Then, to obtain a normal proof it is sufficient to remove all the idle formulae from the tree.

6. The tractability of depth-bounded deduction

The generalized SFP of intelim proofs and refutations paves the way for a feasible decision procedure for intelim deducibility and refutability. The following proposition is proven in [21].

Proposition 6.1. *Whether or not $X \vdash_0 \varphi$ (X is 0-depth refutable) can be decided in time $O(n^2)$ where n is the total number of occurrences of symbols in $X \cup \{\varphi\}$ (in X).*

Proposition 6.1 suggests that the explosivity of 0-depth consequence is far less serious a problem then the explosivity of classical consequence. For, we can always feasibly detect that our premises are 0-depth inconsistent and, therefore, we may as well abstain from drawing bizarre conclusions on their basis. Unlike hidden classical inconsistencies, that may be hard to discover even for agents equipped with powerful (but still bounded) computational resources, 0-depth inconsistency lies, as it were, on the surface. So, we always have a feasible means to ensure that our premises are 0-depth consistent, in which case the consequence relation $\vDash_0$ is *not* explosive, even if these premises are classically inconsistent.

Given Proposition 6.1, a simple analysis shows that, for each $f \in \mathcal{F}$ and each fixed k, $\vdash_k^f$ admits of a feasible decision procedure:

Proposition 6.2. *For each $f \in \mathcal{F}$ and each $k \in \mathbb{N}$, whether or not $X \vdash_k^f \varphi$ ($X \vdash_k^f$), can be decided in polynomial time.*

More precisely, when $f \trianglelefteq \mathtt{sub}$, the complexity of the decision problem is $O(n^{k+2})$, where n is the total number of occurrences of symbols in $X \cup \{\varphi\}$ (in X). In general, the complexity is $O(p(n)^{k+2})$ where p is a polynomial depending on f (recall that the virtual space is, by definition, polynomially bounded).

For unbounded k, the method of intelim trees is a proof system for full classical propositional logic that enjoys the SFP. However, this presentation of classical logic allows also for representing proofs that do not have the SFP simply by permitting applications of PB to formulae that are not subformulae either of the premises or of the conclusion. On the connection between the rule PB and the cut rule of Gentzen's sequent calculus, as well as on the advantages of such a cut-based formalization of classical logic, see [14, 15]. Moreover, for unbounded k, the introduction rules become redundant, since they can be easily derived from the elimination rules with the help of PB. The system consisting only of the *elimination rules* plus PB (with no depth bound) is a complete refutation system for classical propositional logic that enjoys the SFP, since the applications of PB can be restricted to subformulae and the elimination rules obviously preserve the SFP. This system, known as KE, was originally proposed as a more efficent alternative to Smullyan's semantic tableaux. It was shown that KE has an exponential speed-up on semantic tableaux and on Gentzen's cut-free sequent calculus even if we consider its "analytic restriction" that yields only refutations with the SFP [14, 25, 15]. The unbounded method of intelim trees can be seen as an extension of KE, obtained by adding suitable introduction rules for the logical operators. So, intended as a method for full classical propositional logic, intelim trees still have an exponential speed-up on Smullyan's semantic tableaux and on cut-free sequent proofs even when we consider only normal intelim proofs and refutations.

7. Conclusions and further work

The relations $\vDash_k$ and $\vdash_k$ provide an infinite sequence of tractable depth-bounded approximations to classical propositional logic. Observe that in our approach, the tractability of each approximation results from a notion of depth that *applies to single proofs and refutations*. This measure is not based on computational complexity, but on the distinction between actual and virtual information, the tractability of k-depth consequence being derivative. Thus, depth-bounded deduction offers a solution to the *problem of logical omniscience* that appears to overcome the main objection of [44] against complexity-based approaches.[26]

The method of intelim trees combines features of Natural Deduction (it is based on introduction and elimination rules that satisfy a form of the inversion principle [16]) and of Smullyan's Semantic Tableaux (it is a tree method with no discharge rules that can be used as a refutation system as well as proof system), but is essentially more efficient than both. It appears to be heuristically interesting for further developments in a variety of areas. Possible extensions of the work presented here, that may be of interest for researchers in computer science and artificial intelligence, include:

- providing alternative characterizations of classes of inferences whose validity can (or cannot) be shown at a given depth k;
- extending the notions of depth-bounded consequence and depth-bounded inconsistency to non-classical logics by relativizing the primary semantic notions of informational truth and informational falsity to points of some structured space (e.g., possible worlds, information states, etc., equipped with an accessibility relation);[27]
- investigating depth-bounded approximations for the logics of formal inconsistency [12] and, more in general, for paraconsistent logics [45].

Appendix

In this appendix we describe a variant of the method of intelim trees that highlights its analogies and dissimilarities with Gentzen style natural deduction.

[26] "The issue of computational complexity can only make sense for an infinite family of questions, whose answers may be undecidable or at least not in polytime. But for individual questions whose answers we do not know, the appeal to computational complexity misses the issue." [44, p. 462].

[27] From the proof-theoretical point of view, this involves shifting from intelim trees of S-formulae to intelim trees of *labelled S-formulae*. For example, depth-bounded systems for intuitionistic or substructural logics could possibly be developed by adding suitable introduction rules to the KE-like systems discussed in [23] and bounding the applications of the generalized rule of bivalence. A similar generalized rule of bivalence has been also fruitfully used in the context of many-valued logics; see for example [37, 11].

INTRODUCTION RULES

$$\frac{A \quad B}{A \wedge B}\wedge\mathcal{I} \qquad \frac{\neg A}{\neg(A \wedge B)}\neg\wedge\mathcal{I}1 \qquad \frac{\neg B}{\neg(A \wedge B)}\neg\wedge\mathcal{I}2$$

$$\frac{\neg A \quad \neg B}{\neg(A \vee B)}\neg\vee\mathcal{I} \qquad \frac{A}{A \vee B}\vee\mathcal{I}1 \qquad \frac{B}{A \vee B}\vee\mathcal{I}2$$

$$\frac{A \quad \neg B}{\neg(A \to B)}\neg\to\mathcal{I} \qquad \frac{\neg A}{A \to B}\to\mathcal{I}1 \qquad \frac{B}{A \to B}\to\mathcal{I}2$$

$$\frac{A}{\neg\neg A}\neg\neg\mathcal{I} \qquad \frac{A \quad \neg A}{\curlywedge}\curlywedge\mathcal{I}$$

ELIMINATION RULES

$$\frac{A \vee B \quad \neg A}{B}\vee\mathcal{E}1 \qquad \frac{A \vee B \quad \neg B}{A}\vee\mathcal{E}2 \qquad \frac{\neg(A \vee B)}{\neg A}\neg\vee\mathcal{E}1 \qquad \frac{\neg(A \vee B)}{\neg B}\neg\vee\mathcal{E}2$$

$$\frac{\neg(A \wedge B) \quad A}{\neg B}\neg\wedge\mathcal{E}1 \qquad \frac{\neg(A \wedge B) \quad B}{\neg A}\neg\wedge\mathcal{E}2 \qquad \frac{A \wedge B}{A}\wedge\mathcal{E}1 \qquad \frac{A \wedge B}{B}\wedge\mathcal{E}2$$

$$\frac{A \to B \quad A}{B}\to\mathcal{E}1 \qquad \frac{A \to B \quad \neg B}{\neg A}\to\mathcal{E}2 \qquad \frac{\neg(A \to B)}{A}\neg\to\mathcal{E}1 \qquad \frac{\neg(A \to B)}{\neg B}\neg\to\mathcal{E}2$$

$$\frac{\neg\neg A}{A}\neg\neg\mathcal{E} \qquad \frac{\curlywedge}{A}\curlywedge\mathcal{E}$$

Table 4: The classical intelim rules.

We deal with *unsigned formulae* and assume that the propositional language contains also the logical constant $\curlywedge$, denoting "the falsum", intended as an *absurd proposition*. Unlike the intelim rules of standard natural deduction, our intelim rules contain *no discharge rules*. As a result they are not complete for full Boolean logic, but only for the 0-depth logic discussed in Section 2.

To obtain a complete set of rules it is sufficient to add a *single* discharge rule that corresponds to PB: if we have a deduction Π_1 of B from assumptions $\Gamma \cup \{A\}$ and a deduction Π_2 of B from assumptions $\Delta \cup \{\neg A\}$, we thereby have a deduction of B from $\Gamma \cup \Delta$. Schematically:

$$\frac{\begin{array}{c}\Gamma, [A]\\ \Pi_1\\ B\end{array} \qquad \begin{array}{c}\Delta, [\neg A]\\ \Pi_2\\ B\end{array}}{B}\,\mathrm{RB}$$

where the conclusion B does not depend on the "discharged" assumptions A and $\neg A$ that are enclosed in square brackets and represent virtual assumptions. We leave it to the reader to show that Gentzen's rules can be simulated by means of the rules in Table 4 and RB.

If we allow unbounded applications of RB, a *classical intelim deduction* of A from Γ is simply a tree of formulae built in accordance with the intelim rules and RB, such

$$
\begin{prooftree}
\AxiomC{$[p]$}
\AxiomC{$p \to \neg q$}
\RightLabel{}
\BinaryInfC{$\neg q$}
\AxiomC{$q \vee r$}
\RightLabel{}
\BinaryInfC{r}
\AxiomC{$\neg(r \wedge \neg q)$}
\RightLabel{}
\BinaryInfC{$\neg\neg q$}
\RightLabel{}
\UnaryInfC{q}
\AxiomC{$[p]$}
\AxiomC{$p \to \neg q$}
\RightLabel{}
\BinaryInfC{$\neg q$}
\RightLabel{}
\BinaryInfC{$\curlywedge$}
\RightLabel{}
\UnaryInfC{v}
\AxiomC{$\neg v \to s$}
\AxiomC{$t \vee u \to \neg s$}
\AxiomC{$[\neg p]$}
\AxiomC{$p \vee t$}
\RightLabel{}
\BinaryInfC{t}
\RightLabel{}
\UnaryInfC{$t \vee u$}
\RightLabel{}
\BinaryInfC{$\neg s$}
\RightLabel{}
\BinaryInfC{$\neg\neg v$}
\RightLabel{}
\UnaryInfC{v}
\RightLabel{}
\BinaryInfC{v}
\end{prooftree}
$$

Figure 4: Classical intelim deduction

that A occurs in the root and all the undischarged formulae occurring in the leaves belong to Γ. The tree in Fig. 4 is a classical intelim deduction of v from

$$\{p \to \neg q, q \vee r, \neg(r \wedge \neg q), p \vee t, (t \vee u) \to \neg s, \neg v \to s\}.$$

Notice that the last step is an occurrence of RB that discharges the temporary assumptions p (which occurs twice among the leaves of the left subtree) and $\neg p$ (which occurs once among the leaves of the right subtree). This format of intelim trees, with the conclusion as root and the assumptions as leaves, is more perspicuous, since it allows us to visualize immediately the inner structure of the proof. However, it involves a good deal of redundancy in the representation of arguments. This is apparent from the proof tree in Figure 4, where the derivation of $\neg q$ from the assumptions p and $p \to \neg q$ is repeated twice because the conclusion $\neg q$ is used twice as premise of distinct inference steps. Moreover, the format of the rule of bivalence is not particularly convenient for the transformation of proofs and for the implementation of efficient proof-search algorithms. The format presented in Section 4 provides a more concise representation of arguments and is better suited to algorithmic treatment.

References

[1] A. Avron. Non-deterministic matrices and modular semantics of rules. In J.-Y. Béziau, editor, *Logica Universalis*, pages 155–174. Birkäuser, 2005.

[2] A. Avron. A non-deterministic view of non-classical negation. *Studia Logica*, 80(2/3), 2005.

[3] A. Avron and I. Lev. Canonical propositional Gentzen-type systems. In R. Gore, A. Leitsch, and T. Nipkov, editors, *Automated Reasoning: First Joint International Conference (IJCAR)*, Lecture Notes in Artificial Intelligence, 2083, pages 529–544. Springer, 2001.

[4] A. Avron and I. Lev. Non-deterministic multiple-valued structures. *Journal of Logic and Computation*, 15:241–261, 2005.

[5] A. Avron and A. Zamansky. Non-deterministic semantics for logical systems. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 16 of *vol.*, pages 227–304. Springer Verlag, 2nd edition, 2011.

[6] N. D. Belnap Jr. How a computer should think. In G. Ryle, editor, *Contemporary Aspects of Philosophy*, pages 30–55. Oriel Press, 1976.

[7] N. D. Belnap Jr. A useful four-valued logic. In J. M. Dunn and G. Epstein, editors, *Modern uses of multiple-valued logics*, pages 8–37. Reidel, Dordrecht, 1977.

[8] M. Björk. *Stålmarkc's method for automated theorem proving in first order logic*. PhD thesis, Göteborg University, 2003.

[9] A. Borälv. The industrial success of verification tools based on Stålmarck's method. In O. Grumberg, editor, *Computer Aided Verification*, volume 1254 of *Lecture Notes in Computer Science*, pages 7–10. Springer Verlag, 1997. Proceedings of the 9th International Conference, CAV'97 Haifa, Israel, June 2225, 1997.

[10] M. Cadoli and M. Schaerf. Approximate reasoning and non-omniscient agents. In *TARK '92: Proceedings of the 4th conference on Theoretical aspects of reasoning about knowledge*, pages 169–183, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[11] C. Caleiro, J. Marcos, and M.V. Volpe. Bivalent semantics, generalized compositionality and analytic classic-like tableaux for finite-valued logics. To appear in *Theoretical Computer Science*, 2015.

[12] W. Carnielli, M.E. Coniglio, and J. Marcos. Logics of formal inconsistency. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2 of *vol.*, pages 15–107. Springer-Verlag, 2007.

[13] J.M. Crawford and D.W. Etherington. A non-deterministic semantics for tractable inference. In *AAAI/IAAI*, pages 286–291, 1998.

[14] M. D'Agostino. Investigations into the complexity of some propositional calculi. PRG Technical Monographs 88, Oxford University Computing Laboratory, 1990.

[15] M. D'Agostino. Tableau methods for classical propositional logic. In D'Agostino et al. [24], pages 45–123.

[16] M. D'Agostino. Classical natural deduction. In S.N. Artëmov, H. Barringer, A.S. d'Avila Garcez, L.C. Lamb, and J. Woods, editors, *We Will Show Them! (1)*, pages 429–468. College Publications, 2005.

[17] M. D'Agostino. Tractable depth-bounded logics and the problem of logical omniscience. In F. Montagna and H. Hosni, editors, *Probability, Uncertainty and Rationality*, CRM series, pages 245–275. Springer, 2010.

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

[19] M. D'Agostino. Analytic inference and the informational meaning of the logical operators. *Logique et Analyse*, 227:407–437, 2014.

[20] M. D'Agostino. Informational semantics, non-deterministic matrices and feasible deduction. *Electronic Notes in Theoretical Computer Science*, 305:35–52, 2014.

[21] M. D'Agostino, M. Finger, and D.M. Gabbay. Semantics and proof-theory of Depth-Bounded Boolean Logics. *Theoretical Computer Science*, 480:43–68, 2013.

[22] M. D'Agostino and L. Floridi. The enduring scandal of deduction. Is propositionally logic really uninformative? *Synthese*, 167:271–315, 2009.

[23] M. D'Agostino, D.M. Gabbay, and K. Broda. Tableau methods for substructural logics. In D'Agostino et al. [24], pages 397–468.

[24] M. D'Agostino, D.M. Gabbay, R. Hähnle, and J. Posegga, editors. *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.

[25] M. D'Agostino and M. Mondadori. The taming of the cut. *Journal of Logic and Computation*, 4:285–319, 1994.

[26] M. Dalal. Tractable deduction in knowledge representation systems. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92)*, pages 393–402, Cambridge, Massachussetts, 1992.

[27] M. Dalal. Anytime families of tractable propositional reasoners. In *Proceedings of the Fourth International Symposium on AI and Mathematics (AI/MATH-96)*, pages 42–45, 1996.

[28] M. Dalal. Anytime families of tractable propositional reasoners. *Annals of Mathematics and Artificial Intelligence*, 22:297–318, 1998.

[29] M. Dummett. *The logical basis of metaphysics*. Duckworth, London, 1991.

[30] R. Fagin and J.Y. Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.

[31] M. Finger. Polynomial approximations of full propositional logic via limited bivalence. In *9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes in Artificial Intelligence*, pages 526–538. Springer, 2004.

[32] M. Finger. Towards polynomial approximations of full propositional logic. In A.L.C. Bazzan and S. Labidi, editors, *XVII Brazilian Symposium on Artificial Intel ligence (SBIA 2004)*, volume 3171 of *Lecture Notes in Artificial Intel lingence*, pages 11–20. Springer, 2004.

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

[34] M. Finger and R. Wassermann. Approximate and limited reasoning: Semantics, proof theory, expressivity and control. *Journal of Logic and Computation*, 14(2):179–204, 2004.

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

[36] D.M. Gabbay. What is a logical system? An evolutionary view: 1964–2014. In D.M. Gabbay and J.H. Siekmann, editors, *Handbook of the History of Logic, vol. 9: Computational Logic*, pages 41–132. North-Holland, 2014.

[37] R. Hähnle. Tableaux for many-valued logics. In D'Agostino et al. [24], pages 529–580.

[38] J. Hintikka. *Logic, language games and information. Kantian themes in the philosophy of logic.* Clarendon Press, Oxford, 1973.

[39] T.Q. Klassen, S.A. McIlraith, and H.J. Levesque. Towards tractable inference for resource-bounded agents. In *2015 AAAI Spring Symposium Series*, pages 89–95, 2015.

[40] S.C. Kleene. *Introduction to Metamatehmatics.* North-Holland, Amsterdam, 1952.

[41] K. Konolige. What awareness isn't: a sentential view of implicit and explicit belief. In J.Y. Halpern, editor, *Theoretical aspects of reasoning about knowledge. Proceedings of the 1986 conference*, pages 241–250, Los Altos, 1986. Morgan Kaufmann Publishers.

[42] H.J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the national conference on Artificial Intelligence*, pages 198–202, Houston, Texas, 1984.

[43] F. Massacci. *Efficient Approximate Deduction and an Application to Computer Security.* PhD thesis, Università degli Studi di Roma "La Sapienza", 1998.

[44] R. Parikh. Sentences, belief and logical omniscience, or what does deduction tells us? *The Review of Symbolic Logic*, 1(4):459–476, 2008.

[45] G. Priest. Paraconsistent logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 6, pages 287–393. Kluwer Academic Publishers, Dordrecht, 2nd edition, 2002.

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

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

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

[49] Giacomo Sillari. Models of awareness. In G. Bonanno, W. van der Hoek, and M. Wooldridge, editors, *Logic and the Foundations of Games and Decisions*, pages 209–240. University of Amsterdam, 2008.

[50] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9:67–72, 1979.

[51] G. Stålmarck. A system for determining propositional logic theorems by applying values and rules to triplets that are generated from a formula. Swedish Patent No. 467 076 (approved 1992), U.S. Patent No. 5 276 907 (1994), European Patent No. 0403 454 (1995), 1992.