Many-valuedness meets bivalence: Using logical values in an effective way

Carlos Caleiro[1]*, João Marcos[2]†

[1] *SQIG-Instituto de Telecomunicações, and Dept. Mathematics, IST, TU-Lisbon*
[2] *LoLITA and DIMAp, UFRN, and Institut für Computersprachen E1852, TU-Wien*

In spite of the multiplication of truth-values, a noticeable shade of bivalence lurks behind the canonical notion of entailment that many-valued logics inherit from the 2-valued case. Can this bivalence be somehow used to our advantage? The present note briefly surveys the progress made in the last three decades toward making that theme precise from an abstract point of view and effectively extracting some useful procedures from it, harvesting some of its most favorable crops on the domains of semantics and proof-theory.

Key words: Many-valuedness, truth-functionality, bivalence, proof theory, tableaux.

1 ANTIDOTES FOR 'A MAGNIFICENT CONCEPTUAL DECEIPT'

In a 1976 lecture (cf. [42]), the Polish logician Roman Suszko complained that "after 50 years [of the construction of so-called many-valued logics by Jan Łukasiewicz] we still face an illogical paradise of many truths and falsehoods". The bold philosophical thesis behind such an assertion (cf. [40]), updating and extending Frege's discrimination between the sense and the reference of saturated concepts, was that a sharp distinction should be drawn

* email: ccal@math.ist.utl.pt
† email: jmarcos@dimap.ufrn.br

in between the 'algebraic valuations' of the most usual multi-valued truth-functional logics, and their "genuine definition" in terms of two-valued 'logical valuations' (cf. [41]). Suszko's Thesis, as formulated in [29] and [23], roughly says that "every logic is logically two-valued". To put it like that, however, would result in allowing for circumstances in which it is outright wrong, others in which it is but trivial, and still some others in which it is just useless. To do the Thesis some justice, show how and when it works fine, and to exhibit some nice applications for it, we will need some preparation, to be supplied in the present section.

As customary in the general theory of consequence relations (cf. [44]), a propositional logic $\mathcal{L}$ will be characterized as a collection of formulas $\mathcal{S}$ together with a single-conclusion *consequence relation* $\Vdash$ somehow defined as a subset of $\mathsf{Pow}(\mathcal{S}) \times \mathcal{S}$. Moreover, following Łoś & Suszko's methodological work on sentential logics (cf. [27]), we will assume $\mathcal{S}$ to be freely generated over a denumerable set of atoms $\mathsf{At} = \{p_0, p_1, p_2, \ldots\}$ by the constructors $\mathsf{Ct} = \bigcup_{m \in \mathbb{N}} \mathsf{Ct}_m$, where each Ct_m itself denotes a collection of connectives of arity m. We will call a set of formulas Σ *overcomplete* in $\mathcal{L} = \langle \mathcal{S}, \Vdash \rangle$ in case $\Sigma \Vdash \beta$ for every $\beta \in \mathcal{S}$. Taking advantage of the algebraic character of $\mathcal{S}$, for any given total substitution mapping σ:$\mathsf{At} \longrightarrow \mathcal{S}$ there will of course be a unique endomorphism ε^σ:$\mathcal{S} \longrightarrow \mathcal{S}$ that extends it, and we will assume henceforth that the consequence relations of our logics enjoy the following property of *substitutionality* (a.k.a. 'structurality'):

(L0) $\Gamma \Vdash \alpha$ implies $\varepsilon^\sigma(\Gamma) \Vdash \varepsilon^\sigma(\alpha)$

It will also help in the following to denote by $\mathsf{At}[\Sigma]$ the set of atoms that occur in the construction of a given theory $\Sigma \subseteq \mathcal{S}$.

From a semantical viewpoint, let an interpretation for the formulas in $\mathcal{S}$ be a total *valuation* mapping w:$\mathcal{S} \longrightarrow \mathcal{V}_w$ into a given universe of *truth-values* $\mathcal{V}_w$, and assume that $\mathcal{V}_w$ is partitioned into sets of *designated* values $\mathcal{V}_w^1$ and *undesignated* values $\mathcal{V}_w^0$. A *many-valued* semantics Sem here will be any collection of such valuations. From these elements, local ($\models_w$) and global ($\models_{\mathsf{Sem}}$) consequence relations may then be defined according to a canonical concept of T-*entailment* that sets $\Gamma \models_w \alpha$ iff ($w(\gamma) \in \mathcal{V}^0$, for some $\gamma \in \Gamma$, or $w(\alpha) \in \mathcal{V}^1$), and sets $\Gamma \models_{\mathsf{Sem}} \alpha$ iff ($\Gamma \models_w \alpha$, for every $w \in \mathsf{Sem}$). It has since long been known that a consequence relation $\Vdash$ over $\mathcal{S}$ can be characterized by an adequate T-entailment relation $\models_{\mathsf{Sem}}$ iff it enjoys the following properties, for arbitrary $\Gamma \cup \Delta \cup \{\alpha\} \subseteq \mathcal{S}$:

(L1) $\Gamma \cup \{\alpha\} \Vdash \alpha$

(L2) $\Gamma \Vdash \alpha$ implies $\Delta \cup \Gamma \Vdash \alpha$

(L3) ($\Gamma \Vdash \delta$, for every $\delta \in \Delta$, and $\Delta \Vdash \alpha$) imply $\Gamma \Vdash \alpha$

Let us call this result W-*theorem* (cf. [44]). Consider now the set of 'logical' values $\mathcal{V}_2 = \{F, T\}$ such that $\mathcal{V}_2^0 = \{F\}$ and $\mathcal{V}_2^1 = \{T\}$, and for each mapping $w{:}\mathcal{S} \longrightarrow \mathcal{V}$ let its *bivalent counterpart* $b_w{:}\mathcal{S} \longrightarrow \mathcal{V}_2$ be defined by setting $b_w(\varphi) = F$ if $w(\varphi) \in \mathcal{V}^0$ and $b_w(\varphi) = T$ if $w(\varphi) \in \mathcal{V}^1$. Collecting all such bivalent valuations, hereon referred to as *bivaluations*, into Sem_2, it is obvious that $\Gamma \models_{\mathsf{Sem}} \alpha$ iff $\Gamma \models_{\mathsf{Sem}_2} \alpha$. This may be said to constitute the very core of Suszko's observation on logical 2-valuedness, and we will call this result S-*theorem*. A brief review of the above mentioned theorems and their proofs (also in the multiple-conclusion case) can be found in [31].

A particularly interesting genre of many-valued semantics Sem is obtained when one fixes the sets $\mathcal{V}_w$ and $\mathcal{V}_w^1$ (call them $\mathcal{V}$ and $\mathcal{V}^1$), for every $w \in \mathsf{Sem}$, and also fixes the interpretation $\lceil © \rceil$ of each $© \in \mathsf{Ct}_m$ in such a way that, for every $w \in \mathsf{Sem}$ and $\alpha_1, \ldots, \alpha_m \in \mathcal{S}$, the following equation holds good:

(S1) $w(©(\alpha_1, \ldots, \alpha_m)) = \lceil © \rceil(w(\alpha_1), \ldots, w(\alpha_m))$

This means that we may think now of the universe of truth-values $\mathcal{V}$ as organized in terms of an algebra with the same similarity type of the algebra of formulas, where to each syntactical constructor $©{:}\mathcal{S}^m \longrightarrow \mathcal{S}$ there corresponds a semantical operator $\lceil © \rceil{:}\mathcal{V}^m \longrightarrow \mathcal{V}$. This also means, of course, that any basic *state of affairs* given by a total mapping $e{:}\mathsf{At} \longrightarrow \mathcal{V}$ can be uniquely extended into a homomorphic valuation $w^e{:}\mathcal{S} \longrightarrow \mathcal{V}$ from the algebra of formulas into the algebra of truth-values. Any semantics given by the collection Hom of all such homomorphisms is called *truth-functional*. Now, say that the sets of formulas Σ and Π are *disconnected* in case $\mathsf{At}[\Sigma] \cap \mathsf{At}[\Pi] = \varnothing$. A remarkable result by Shoesmith & Smiley (cf. [38]) shows that a logic $\mathcal{L}$ is characterized by a truth-functional T-entailment iff it enjoys all the (L#)-properties above, plus the following *cancellation* property:

(L4) $\bigcup_{k \in K} \Gamma_k \cup \Gamma \Vdash \varphi$ implies $\Gamma \Vdash \varphi$, once, for every $k \in K$, we have that $\Gamma \cup \{\varphi\}$ and Γ_k are disconnected, and that Γ_k is not overcomplete

A logic $\mathcal{L}$ is said to be *genuinely κ-valued* if κ is the cardinality of the smallest collection of truth-values $\mathcal{V}_\kappa$ with the help of which $\mathcal{L}$ can be given a truth-functional semantics. The drama set up by the S-theorem reaches its climax exactly in the cases in which $\mathcal{L}$ turns out to be genuinely κ-valued, for some $\kappa > 2$: in such a case a bivalent characterization of $\mathcal{L}$ will presume an open abandonment of a truth-functional characterization.

A genuinely κ-valued logic $\mathcal{L}$ with a set of constructors Ct is said to be *functionally complete* in case any operator $\lceil\circledast\rceil$ over $\mathcal{V}_\kappa$ can be defined by way of some convenient combination of operators associated to the constructors Ct. Consider any two distinct values $v_i, v_j \in \mathcal{V}_\kappa$, let θ^{ij} be such that $\mathsf{At}[\{\theta^{ij}\}] = \{p_0\}$ and let $\sigma_{[p_n \mapsto \delta]}$ be a substitution mapping that outputs the value δ with input p_n and behaves as the identity mapping otherwise. Given a state of affairs e such that $e(p_i) = v_i$ and $e(p_j) = v_j$, and its corresponding valuation w^e, we say that the formula θ^{ij} *effectively separates* v_i and v_j in case $b_{w^e}(\varepsilon^\sigma_{[p_0 \mapsto p_i]}(\theta^{ij})) \neq b_{w^e}(\varepsilon^\sigma_{[p_0 \mapsto p_j]}(\theta^{ij}))$. Obviously, it suffices to take θ^{ij} as p_0 itself to separate truth-values that are not both designated, nor both undesignated. For pairs of values from the same partition class, however, it may or it may not be the case that the logic $\mathcal{L}$ has the linguistic resources to separate them. We will here say that a genuinely κ-valued logic $\mathcal{L}$ is *sufficiently expressive* when its language is expressive enough to separate each pair of truth-values from the collection $\mathcal{V}_\kappa$. Clearly, functional completeness gives a sufficient, yet not necessary, condition for a logic to be sufficiently expressive. Noticeably, for any genuinely κ-valued logic $\mathcal{L}$ with $\kappa > 2$, either $\mathcal{L}$ or some conservative extension of $\mathcal{L}$ is bound to be sufficiently expressive⋆ . A full proof of this fact will appear in [18], but an illustration can easily be drawn at the light of the theory of logical matrices (see [44]). If $\mathcal{L}$ is genuinely κ-valued then the Leibniz congruence [9] of its κ-valued semantics must be the identity. Thus, in order to separate two given truth-values v_i and v_j, it is sufficient to note that the congruence generated by the equation $v_i \approx v_j$ is incompatible with the distinction between designated and undesignated values. Concretely, there must exist a formula $\varphi(p_0, p_1, \ldots, p_m) \in \mathcal{S}$ and values $v_{t_1}, \ldots, v_{t_m} \in \mathcal{V}_\kappa$ such that $w^0(\varphi) \in \mathcal{V}^0_\kappa$ and $w^1(\varphi) \in \mathcal{V}^1_\kappa$, where $w^0(p_0) = v_i$, $w^1(p_0) = v_j$ and $w^0(p_n) = w^1(p_n) = v_{t_n}$ for each $n \in \{1, \ldots, m\}$. Hence, by extending the syntax of the logic with 0-ary constructors $\boxtimes_{t_1}, \ldots, \boxtimes_{t_m}$, and working in the conservative extension of $\mathcal{L}$ obtained by requiring that each $\lceil\boxtimes_{t_n}\rceil = v_{t_n}$, it is simple to see that the envisaged separating formula θ^{ij} can be set to be $\varphi(p_0, \boxtimes_{t_1}, \ldots, \boxtimes_{t_m})$.

Back from semantics to abstract properties of consequence relations, given a logic $\mathcal{L} = \langle \mathcal{S}, \Vdash \rangle$, we say that two formulas γ and δ are *$\mathcal{L}$-equivalent*,

---

⋆ More recently, a similar requirement on 'expressiveness' has also been employed in [2], with the goal of providing a sufficient condition for the extraction of adequate *n-sided sequents* for logics based on an extended notion of truth-functionality. It should be stressed that our homonymous notion, nonetheless, as will be illustrated below, always aimed at providing a sufficient condition for the extraction of adequate *two-valued semantics* for broadly truth-functional logics, and the consequent extraction of adequate *two-signed proof systems* for such logics.

and denote this by $\gamma \equiv_{\mathcal{L}} \delta$, if both $\{\gamma\} \Vdash \delta$ and $\{\delta\} \Vdash \gamma$. An important feature of classical logic, shared also by all the usual modal logics, is given by the enjoyment of the so-called *replacement property*, according to which equivalent formulas are 'logically indistinguishable', that is:

(L5) $\alpha \equiv_{\mathcal{L}} \beta$ implies $\varepsilon_{[q\mapsto\alpha]}(\varphi) \equiv_{\mathcal{L}} \varepsilon_{[q\mapsto\beta]}(\varphi)$,
for any $\varphi \in \mathcal{S}$ and any $q \in \mathsf{At}$

Suszko sometimes called this property the 'Fregean Axiom' (cf. [40,42]) and claimed that "the construction of [the] so-called many-valued logics by Jan Łukasiewicz was the effective abolition of the Fregean Axiom". However, it is worth noticing that such a claim is only true, in fact, for sufficiently expressive logics. There are indeed genuinely κ-valued logics, with $\kappa > 2$, that enjoy the replacement property: a simple example would be that of a truth-functional logic $\mathcal{L}^{\circledast}$ with $\mathcal{V} = \{v_0, v_1, v_2\}$, $\mathcal{V}^1 = \{v_2\}$, and a single binary constructor $\circledast$ interpreted by setting $\lceil\circledast\rceil(v_n, v_n) = v_n$, for $n \in \{0, 1, 2\}$, and $\lceil\circledast\rceil(v_i, v_j) = v_2$, otherwise. A corrected version of Suszko's claim should then be something like: "a sufficiently expressive truth-functional logic may only satisfy the replacement property in case it is genuinely 2-valued". A sufficiently expressive conservative extension of $\mathcal{L}^{\circledast}$ could be obtained for instance by adding to the language of this logic a 0-ary constructor $\boxtimes$ interpreted by setting $\lceil\boxtimes\rceil = v_0$, but then of course this logic would fail the replacement property (notice how (L5) fails if one considers, e.g., $\alpha = p_0$, $\beta = p_0 \circledast \boxtimes$ and $\varphi = q \circledast p_0$). Once we will be interested below exclusively on sufficiently expressive many-valued logics, all the non-classical truth-functional logics we will consider are indeed to fail replacement — and this fact would certainly gratify Suszko in his analysis of the Fregean Axiom.

Several other important aspects of truth-functionality are discussed in [33], where open problems related to 'computationally well-behaved' generalizations of the notion of truth-functionality are also mentioned. An interesting non-deterministic variety of truth-functionality has been proposed in [3], where again the sets of truth-values are fixed for all interpretation mappings, but this time for each $\copyright \in \mathsf{Ct}_m$ there corresponds an operator $\lfloor\copyright\rfloor : \mathcal{V}^m \longrightarrow \mathsf{Pow}(\mathcal{V}) \setminus \varnothing$ such that, for every $w \in \mathsf{Sem}$ and formulas $\alpha_1, \ldots, \alpha_m$:

(S2) $w(\copyright(\alpha_1, \ldots, \alpha_m)) \in \lfloor\copyright\rfloor(w(\alpha_1), \ldots, w(\alpha_m))$

This means that there might be a number of ways of interpreting the meaning of each constructor as applied to a given tuple of inputs. Consider for instance the simple example of a logic having a binary constructor $\supset$ interpreted deterministically over $\mathcal{V}_2 = \{F, T\}$, $\mathcal{V}_2^0 = \{F\}$, and $\mathcal{V}_2^1 = \{T\}$ as the classical implication, that is, such that $v_1\lfloor\supset\rfloor v_2 \in \mathcal{V}_2^0$ iff ($v_1 \in \mathcal{V}_2^1$ and $v_2 \in \mathcal{V}_2^0$), and having a 0-ary constructor $\mathbb{I}$ interpreted non-deterministically by setting $\lfloor\mathbb{I}\rfloor = \mathcal{V}_2$. In that case the resulting logic would not enjoy property (L4) (just consider $K = \{\star\}$, $\Gamma_\star = \{p_0 \supset \mathbb{I}, p_0\}$, $\Gamma = \varnothing$ and $\varphi = p_1 \supset \mathbb{I}$), and would fail thus to be truth-functional (cf. [33]). It is not entirely clear, however, what the *meaning* of Suszko's Thesis on logical two-valuedness would be in such a scenario, and in particular it is not as yet known how the class of consequence relations related to such a wider class of non-deterministically truth-functional logics is to be characterized from an abstract viewpoint.

The next sections will show how logical two-valuedness has been explored from a constructive perspective. To be perfectly fair, however, we will end the present section by briefly mentioning some ways in which a logic may fail to be bivalent, even in the sense of the S-theorem. The obvious way of obtaining that effect, of course, would be by proposing consequence relations that fail some of the (L#)-properties. Such is the case of the notion of 'inferential many-valuedness' studied in [30], that goes against Suszko's Thesis in that it turns out to be based on 'logical three-valuedness' and on a slightly modified notion of entailment. Another illuminating way of eluding the bivalence behind the notion of T-entailment would be by allowing either $\mathcal{V}^1 \cap \mathcal{V}^0$ or $\mathcal{V} \setminus (\mathcal{V}^1 \cup \mathcal{V}^0)$ to be non-empty, as proposed in [43].

2 THE EXTRACTION OF BIVALENT SEMANTICS FOR FINITE-VALUED LOGICS

The use of bivalent non-truth-functional semantics has proven extremely useful in the domain of non-classical logics, especially when no other insightful varieties of semantics are available for those logics, at a given moment. The realms of paraconsistent and paracomplete logics, for instance, have indeed benefitted a lot from the bivalent approach (cf. [26]), in particular when one is dealing with logics that fail the replacement property and also fail to have genuinely finite-valued semantics (cf. [21]). The pre-requisites for obtaining completeness for such bivalent semantics are now well understood (cf. [7]), and associated decidability procedures known as 'quasi matrices' have been used since [22]. Such procedures are in fact available, as we have argued in [11], at least when the bivalent semantics is presented in a certain specific 'dyadic' format.

Suszko's Thesis, however, is equally valid when the logics *do* have a finite-valued truth-functional semantics, and this section and the next will discuss the worthiness of the Thesis also for such a domain. We will start here by succinctly appraising the more recent efforts toward *constructively* securing, prêt-à-porter, the finest consequences of the S-theorem.

One of the first announcements concerning the availability of a bivalent semantics for a genuinely 3-valued logic, Łukasiewicz's logic Ł$_3$, can be found in [41] — though the corresponding clauses concerning the collection of bivalent interpretation mappings appear only in [28]. One cannot exaggerate in asserting, however, that that specific adequate bivalent characterization for Ł$_3$, however, looked rather mystifying, as no effort was made to clarify how it could be obtained directly from the set of truth-tables that characterize the original semantics of the logic. Given the considerably non-constructive character of the S-theorem, nonetheless, the definition of an effective procedure for obtaining such a bivalent characterization should be particularly welcomed. A substantial step toward that goal was made in [6], where the author suggested that in many cases an 'algebraic' truth-value can be constructively exchanged by a unique 'binary print', in terms of a tuple of values from $\mathcal{V}_2$, with the exclusive help of the original linguistic resources of the given logic. To make matters more concrete, from this point on we will illustrate the mentioned ideas and procedures by way of the $\{\neg, \rightarrow\}$-fragment of the Gödel logic G_3, the first of a well-known hierarchy of many-valued logics that approximate intuitionistic logic from above. In G_3 we have the values $\mathcal{V} = \{v_0, v_1, v_2\}$, naturally ordered by their corresponding indices; we also have $\mathcal{V}^1 = \{v_2\}$, and sometimes abbreviate v_0 by f and v_2 by t. The operators of G_3 are defined by setting (along the lines of equation (S1), from the previous section): $\lceil\neg\rceil(v) = t$ if v is f, and $\lceil\neg\rceil(v) = f$ otherwise; $\lceil\rightarrow\rceil(v_i, v_j) = t$ if $i \leq j$, and $\lceil\rightarrow\rceil(v_i, v_j) = v_j$ otherwise. The valuations $w{:}\mathcal{S} \longrightarrow \mathcal{V}$ in Hom^{G_3} are all the mappings that respect the above restrictions on the meaning of the operators. Now, it is easy to see that the formula $\neg p_0$ effectively separates the undesignated values f and v_1, which would otherwise both be mapped into the logical value F. Accordingly, one might think of rewriting the initial algebraic values of G_3 in terms of their corresponding uniquely identifying binary prints $\langle\lceil p_0\rceil, \lceil\neg p_0\rceil\rangle$, to the effect that: $\mathcal{V}_B^0 = \{\langle F, T\rangle, \langle F, F\rangle\}$ and $\mathcal{V}_B^1 = \{\langle T, F\rangle\}$. Notice, however, that for our goal of uniformly expressing the algebraic valuations $w{:}\mathcal{S} \longrightarrow \mathcal{V}$ in terms of logical (bi)valuations $b_w{:}\mathcal{S} \longrightarrow \mathcal{V}_2$, we are still one step short: what we have at this point are algebraic valuations in disguise, of the form $w_B{:}\mathcal{S} \longrightarrow \mathcal{V}_B$, which just exchange the initial many-valued codomain for the corresponding tuples in terms of binary prints.

A fuller study of how such a procedure can realize Suszko's Thesis and smoothly fit into the variegated many-valued scenarios from the literature was presented in [14] and [13]. Subsequently, in a number of papers starting with [11] we have finally shown how Suszko's two-valued reduction can be fully accomplished, in a constructive way. In particular, we have also proposed a procedure for extracting the axioms on the class of bivaluations that correspond to a given finite-valued logic. The input of that first algorithm corresponds to the specification of a sufficiently expressive genuinely κ-valued logic, and its output are the clauses of a sound and complete bivalent semantics for it. The basic idea, to be sure, is still to use the available linguistic resources to produce the effective separation of each pair of truth-values, and then use the corresponding syntactically expressed binary prints of those values to couch the original many-valued specification into a two-valued environment. Going back to the above illustration, that of the logic G_3, the rough idea is to directly use the tuple $\langle\varphi, \neg\varphi\rangle$, instead of its corresponding interpretation, whenever we need to refer to a formula φ. The truth-tables of G_3 could then be exhaustively described, in principle, by stating convenient restrictions governing the bivaluations assigning values T and F to φ and $\neg\varphi$, when φ is matched either to $\neg\alpha$ or to $\alpha\rightarrow\beta$.

The whole idea can be illustrated by defining the bivalent semantics of G_3, where we employ a classical metalinguistic notation in which a ',' replaces an *and*, a '|' replaces an *or*, a '$\Longrightarrow$' stands for an *if-then* assertion, and a '$\divideontimes$' symbol represents the *absurd*. To start with, we postulate

$\mathsf{biv}[T0]$		$\Longrightarrow$	$(b(\alpha) = F \mid b(\alpha) = T)$
$\mathsf{biv}[C0]$	$(b(\alpha) = F\,,\, b(\alpha) = T)$	$\Longrightarrow$	$\divideontimes$
$\mathsf{biv}[C1]$	$(b(\alpha) = T\,,\, b(\neg\alpha) = T)$	$\Longrightarrow$	$\divideontimes$

On the one hand, axioms $\mathsf{biv}[T0]$ and $\mathsf{biv}[C0]$ follow from the definition of b_w and the fact that each w is a total function. On the other hand, $\mathsf{biv}[C1]$ reflects the semantically unobtainable assignment, given the meaning of $\neg$ in G_3, that would try to force $w(\varphi) = t = w(\neg\varphi)$.

Extracting convenient clauses governing the whole bivaluation semantics for the implication connective of G_3 is also simple. Consider, for instance, the implication of G_3. A brief analysis of its semantics shows that $\alpha \rightarrow \beta$ is 'false' according to te bivalent setting provided by the S-theorem precisely when its value is v_0 or v_1, which amounts to requiring that the value of α is bigger than the value of β, that is, $\langle\alpha, \beta\rangle$ are assigned either the values $\langle v_1, v_0\rangle$ or $\langle v_2, v_0\rangle$ or $\langle v_2, v_1\rangle$. Using the correspondence with the binary prints mentioned above, one could write:

$$\mathsf{biv}[\to]\langle F\rangle^\star \qquad b(\alpha\to\beta)=F \implies \begin{array}{l} (b(\alpha)=F\,,\,b(\neg\alpha)=F\,,\,b(\beta)=F\,,\,b(\neg\beta)=T) \\ \mid\ (b(\alpha)=T\,,\,b(\neg\alpha)=F\,,\,b(\beta)=F\,,\,b(\neg\beta)=T) \\ \mid\ (b(\alpha)=T\,,\,b(\neg\alpha)=F\,,\,b(\beta)=F\,,\,b(\neg\beta)=F) \end{array}$$

Repeating this method, we can obtain a complete characterization of the bivaluation semantics of G_3. Note, however, that the descriptions obtained can be greatly simplified if one uses the classical metalanguage to manipulate the bivaluation axioms so as to reduce their inner redundancies. For instance, one never needs to write both $b(\varphi)=T$ and $b(\neg\varphi)=F$, as the latter expression, $b(\neg\varphi)=F$, follows from the former, $b(\varphi)=T$,' in the presence of $\mathsf{biv}[C1]$. Using usual classical equivalences to reduce thus the overall complexity of the expression in disjunctive normal form that appears in the right of each meta-implication $\implies$, a much simpler way of defining the *same* collection of bivaluations is at hand. In particular, $\mathsf{biv}[\to]^\star$ simplifies to:

$$\mathsf{biv}[\to]\langle F\rangle \qquad b(\alpha\to\beta)=F \implies (b(\neg\alpha)=F\,,\,b(\neg\beta)=T)\ \mid\ (b(\alpha)=T\,,\,b(\beta)=F)$$

The full list of simplified axioms that should be respected by each $b\in\mathsf{Sem}_2^{G_3}$ results from adding to $\mathsf{biv}[T0]$, $\mathsf{biv}[C0]$, $\mathsf{biv}[C1]$ and $\mathsf{biv}[\to]\langle F\rangle$ the following conditions:

$$\mathsf{biv}[\to]\langle T\rangle \qquad b(\alpha\to\beta)=T \implies (b(\neg\alpha)=T)\ \mid\ (b(\beta)=T)\ \mid\ (b(\alpha)=F\,,\,b(\neg\beta)=F)$$

$$\mathsf{biv}[\neg\to]\langle F\rangle \qquad b(\neg(\alpha\to\beta))=F \implies (b(\neg\alpha)=T)\ \mid\ (b(\neg\beta)=F)$$

$$\mathsf{biv}[\neg\to]\langle T\rangle \qquad b(\neg(\alpha\to\beta))=T \implies b(\neg\alpha)=F\,,\,b(\neg\beta)=T$$

$$\mathsf{biv}[\neg\neg]\langle F\rangle \qquad b(\neg(\neg\alpha))=F \implies b(\neg\alpha)=T$$

$$\mathsf{biv}[\neg\neg]\langle T\rangle \qquad b(\neg(\neg\alpha))=T \implies b(\neg\alpha)=F$$

One may show $\mathsf{Sem}_2^{G_3}$ to constitute an adequate bivalent semantics for the genuinely 3-valued logic G_3 by way of the two results that follow:

Convenience. Given $w\in\mathsf{Hom}^{G_3}$, define b_w, for every $\varphi\in\mathcal{S}$, by setting $b_w(\varphi)=T$ if $w(\varphi)=t$, and $b_w(\varphi)=F$ otherwise. Then, $b_w\in\mathsf{Sem}_2^{G_3}$.

Proof. Each of the above bivaluation axioms has to be checked against the given definition. We choose here $\mathsf{biv}[\neg\to]\langle F\rangle$ to offer details of a representative case. Accordingly, assume that $b_w(\neg(\alpha\to\beta))=F$. By the definition of b_w, this is the case iff $w(\neg(\alpha\to\beta))\neq t$. Some easy calculation with the meanings of $\neg$ and $\to$ in G_3, however, guarantee that $w(\neg(\alpha\to\beta))\neq t$ iff

(A) $w(\neg\alpha) = t$ or (B) $w(\neg\beta) \neq t$. Using the definition of b_w again, one concludes that (A) or (B) is the case iff either (A′) $b_w(\neg\alpha) = T$ or (B′) $b_w(\neg\beta) = F$. But the disjunction (A′) or (B′) constitutes exactly the scenario allowed by $\mathsf{biv}[\neg\!\rightarrow]\langle F\rangle$. Similar reasoning takes care of the other bivaluation axioms. □

Representability. Given $b \in \mathsf{Sem}_2^{G_3}$, define w_b by setting:

$$\begin{array}{lll} w_b(\varphi) = f & \text{if} & b(\neg\varphi) = T \\ w_b(\varphi) = v_1 & \text{if} & b(\varphi) = b(\neg\varphi) = F \\ w_b(\varphi) = t & \text{if} & b(\varphi) = T \end{array}$$

Then, $w_b \in \mathsf{Hom}^{G_3}$.

Proof. Here one must check that the given definition provides a 3-valued mapping that respects all the restrictions concerning the meaning of the operators of G_3. It is often helpful to notice, given $\mathsf{biv}[C1]$, that $b(\varphi) = T$ implies $b(\neg\varphi) = F$. Now, for the details of a representative case, assume that $w_b(\alpha\!\rightarrow\!\beta) = f$. The definition of w_b says that this is the case exactly when $b(\neg(\alpha\!\rightarrow\!\beta)) = T$. But $\mathsf{biv}[\neg\!\rightarrow]\langle T\rangle$ guarantees that $b(\neg\alpha) = F$ and $b(\neg\beta) = T$. From the definition of w_b, one may conclude from $b(\neg\beta) = T$ alone that $w_b(\beta) = f$, and from $b(\neg\alpha) = F$ that $w_b(\alpha) \neq f$. The conclusions are appropriate, as an inspection of the truth-table of $\rightarrow$ shows that $v_a \lceil\rightarrow\rceil v_b = f$ only if $b = 0$ and $a \in \{1, 2\}$. The verifications for the case of other truth-values and connectives are entirely analogous. □

While in the finite-valued truth-functional case, 3-valued in the case of G_3, a usual decision procedure by way of truth-tables may immediately be associated, it is not at all obvious that to an arbitrary given collection of bivaluations there should be also an associated decision procedure — and in general this is indeed *not* the case. However, an essential feature of the recipe we used for producing the bivaluation axioms is precisely that we retain this key property. In the case of G_3, we can measure the number of necessary evaluation steps by way of the function $\mathsf{dpth} : \mathcal{S} \rightarrow \mathbb{N}$ inductively defined as follows:

$$\begin{array}{l} \mathsf{dpth}(p) = \mathsf{dpth}(\neg p) = 0 \quad \text{if } p \in \mathsf{At} \\ \mathsf{dpth}(\alpha \rightarrow \beta) = 1 + \mathsf{dpth}(\alpha) + \mathsf{dpth}(\neg\alpha) + \mathsf{dpth}(\beta) + \mathsf{dpth}(\neg\beta) \\ \mathsf{dpth}(\neg(\alpha \rightarrow \beta)) = 1 + \mathsf{dpth}(\alpha) + \mathsf{dpth}(\neg\alpha) + \mathsf{dpth}(\beta) + \mathsf{dpth}(\neg\beta) \\ \mathsf{dpth}(\neg(\neg\alpha)) = 1 + \mathsf{dpth}(\alpha) + \mathsf{dpth}(\neg\alpha) \end{array}$$

Effectiveness. Given $\varphi(p_1, \ldots, p_m) \in \mathcal{S}$ and $b \in \mathsf{Sem}_2^{G_3}$, the value $b(\varphi)$ is uniquely determined by the values $b(p_n)$, $b(\neg p_n)$ for $n \in \{1, \ldots, m\}$.

Moreover, $b(\varphi)$ can be computed using at most $\mathsf{dpth}(\varphi)$ applications of the bivaluation axioms.

Proof. The first statement is a consequence of the convenience and representability of $\mathsf{Sem}_2^{G_3}$, as the values $b(p_n)$, $b(\neg p_n)$ for $n \in \{1, \ldots, m\}$ uniquely determine a 3-valuation from which b obtains in a unique way.

The second statement follows easily by induction on the structure of the formula φ, using the bivaluation axioms corresponding to each case, whose right-hand sides are easily seen to comply with the definition of dpth. In the base case, let φ be p, or $\neg p$, for some $p \in \mathsf{At}$. As $b(p)$ and $b(\neg p)$ are given, we are done with $\mathsf{dpth}(p) = \mathsf{dpth}(\neg p) = 0$ applications of the valuation axioms. Regarding the induction step, let us consider the case when φ is of the form $\alpha \to \beta$. By induction hypothesis, the values of $b(\alpha)$, $b(\neg\alpha)$, $b(\beta)$ and $b(\neg\beta)$ can be computed using at most $\mathsf{dpth}(\alpha)$, $\mathsf{dpth}(\neg\alpha)$, $\mathsf{dpth}(\beta)$ and $\mathsf{dpth}(\neg\beta)$ applications of the axioms, respectively. Then, exactly one of the axioms $\mathsf{biv}[\to]\langle T\rangle$ or $\mathsf{biv}[\to]\langle F\rangle$ will apply and yield $b(\alpha \to \beta)$ in $\mathsf{dpth}(\alpha \to \beta)$ steps. The remainder cases for φ, that is, $\neg(\alpha \to \beta)$ and $\neg(\neg\alpha)$ are analogous. □

This ends our illustration concerning G_3. For the case of other logics, the extraction procedure is essentially the same, namely, given a finite-valued logic $\mathcal{L}$ with a primitive collection of operators Op:

(E1) Find a collection Sep of unary formulas that can produce the effective separation of the truth-values; if such a collection is not fully definable from the original linguistic resources of the given logic, conservatively extend the latter by the addition of convenient unary operators. By stipulation, we'll leave the omnipresent identity unary formula id (for which $w(\mathsf{id}(p)) = w(p)$) out of Sep.

(E2) Use the binary prints corresponding to the separation formulas in order to describe a set of restrictive axioms governing the bivaluations. These axioms will include:

- **(E2.1)** an axiom $\mathsf{biv}[©]\langle v\rangle$ for each $© \in \mathsf{Op} \setminus \mathsf{Sep}$, and each $v \in \mathcal{V}_2$;
- **(E2.2)** an axiom $\mathsf{biv}[Ⓢ©]\langle v\rangle$ for each combination of $Ⓢ \in \mathsf{Op}$ and $© \in \mathsf{Op}$, and each $v \in \mathcal{V}_2$;
- **(E2.3)** axioms $\mathsf{biv}[T0]$ and $\mathsf{biv}[C0]$, guaranteeing that each $b \in \mathsf{Sem}_2$ is a total function;

(E2.4) axioms $\mathsf{biv}[Cn]$, for $n > 0$, for each unobtainable bivalent semantic situation, that is, for each situation that does not correspond to the binary print of an algebraic value.

For more formal details on the above described general procedure, and many further illustrations, we had better direct the reader to the appropriate sources: for languages that are not sufficiently expressive and the corresponding conservative extensions that might be necessary to make them expressive enough, the preparatory phase mentioned in step (E1) is described in [18] and in the previous section; descriptions of how the bivaluation axioms in steps (E2.1) and (E2.2) look like were presented in [11] and updated in [16]; the final form of the axioms in step (E2.4) can be found in [34] and [18]. Analogously to what we have done above using the non-canonical complexity measure dpth for G_3, in [16, 18] we also show how, in the general finite-valued case, a well-founded evaluation order supporting the effectiveness of the bivalent semantics can also be obtained alongside the extraction procedure.

The next section will show an immediate application of our effective version of the S-theorem. The new procedure will consist in associating adequate analytic proof systems to a given bivaluation semantics. Such proof systems, that will here be presented as classic-like tableau systems, are available not just for the case of bivaluation axioms obtainable from one of our above mentioned extraction algorithms, but in general for any collection of bivaluation axioms that are formulated in a very general format that will be briefly discussed below.

3 A BIRD'S EYE VIEW OF SOME APPLICATIONS TO MODEL THEORY AND TO PROOF THEORY

A number of applications may be envisaged for bivalent semantics, some of which we will briefly examine in this section. One of their most striking advantages, at first sight, lies in providing a uniform classic-like framework in which a plethora of different non-classical logics can be specified, and more easily compared with each other. We shall insist on this point below.

From a model-theoretic viewpoint, besides helping in establishing decidability for a large class of non-classical logics, another productive application for a bivalent semantics consists in providing a useful intermediary step in the process of associating another more informative kind of semantics to the same logic. Such has been the case, for instance, with the use of bivalent semantics in the proof of completeness of a certain semantics given by way of *combinations* of finite-valued truth-functional scenarios, even when the given non-classical logic turn out *not* to be characterizable by way of a genuinely finite-valued truth-functional semantics (cf. [32]). The underlying idea is somehow to 'split' a given complex logic in terms of more well-behaved ingredients (cf. [31]), a very generally applicable approach to model theory known as *possible-translations semantics*, first proposed in [19].

Now, for the case of logics that *do* have a finite-valued truth-functional semantics, the constructive procedure for extracting a bivalent characterization for them, reported upon in the previous section, has borne some fruits also from a proof-theoretical perspective. Even though general axiomatization algorithms for finite-valued logics have been known for long, they are typically based on indiscriminate extensions of the linguistic resources of the original logics, as in [39], or else they produce rules, as in [25] and [5], that do not easily lend themselves to the comparison of a genuinely κ_a-valued to a genuinely κ_b-valued logic, when $\kappa_a \neq \kappa_b$. Such a general non-uniform approach to finite-valued logics in terms of *tableaux*, for example, has been available at least since [20]. On what concerns the comparison between the inferences sanctioned by two different logics, obvious difficulties arise if these logics are specified over different languages, as this might require quite some ingenuity in finding suitable ways of translating assertions from one logic to another. The above mentioned 'traditional' methods for extracting adequate collections of tableau rules for a given many-valued logic typically meets the same difficulty, but at a different level: even without modifying the object language of a given logic, in transforming the truth-values of such logic (or collections of such truth-values) into signs to be put in front of the formulas, such methods may very easily, again, introduce new differences that make rules from different logics hard to compare. For all such cases, thus, it would seem that the introduction of more uniform frameworks, such as the those we illustrate here, could only help for the logic comparison task.

In the finite-valued case, at any rate, a novel conservative algorithm has been proposed (cf. [11]) that produces tableau rules with only two labels, as in the classical case, exactly by exploring the underlying bivalence behind the notion of T-entailment, as supplied by our constructive rendering of the S-theorem. Furthermore, as argued in [35], the uniform classic-like approach, with its emphasis on distinguishing among designated values and among undesignated values, may benefit even the user that wishes to compare the deductive strength of truth-functional logics based on essentially the same algebraic structures, with $\mathsf{Card}(\mathcal{V}_a) = \mathsf{Card}(\mathcal{V}_b)$ yet $\mathsf{Card}(\mathcal{V}_a^0) \neq \mathsf{Card}(\mathcal{V}_b^0)$.

A full implementation of the above mentioned algorithm, receiving as input the specification of a sufficiently expressive finite-valued logic, together with the appropriate separation formulas, and producing as output a complete set of tableau rules as a ready-to-use `Isabelle` theory (cf. [37]) was presented in [36], and made available online† . The tableau theory implemented in the framework of the higher-order metalanguage of a very flexible proof-assistant includes structural rules that allow for the relatively easy derivation, by the user, of theorems and rules of the given logics, as well as for the comparison between different logics, all re-specified now in a uniform two-signed framework. Progress toward the complete automation of the associated proof procedures, however, was initially hindered by the fact that the set of tableau rules produced by the procedure laid out in [11] includes a kind of dual-cut branching rule that in principle would sanction the production of derivations that do not terminate, should the user make some bad choices along their construction. Though it had been known that in general this dual-cut rule was *not* eliminable, a conjecture had been made that all uses of cut in our systems could be made 'analytic', as in [24]. Such non-eliminable use of a dual-cut rule was in fact an ordinary feature of the bivaluation semantics presented in dyadic format, as studied in [11], even for non-finite-valued logics.

Now, instead of proceeding towards directly proving the above mentioned conjecture about analytic cuts, for the finite-valued case, we have later proposed, in [16], a novel algorithm that receives the very same many-valued specifications, and outputs adequate *cut-free* tableau systems. Let's illustrate below how this second algorithm works, again for the case of the logic G_3, as in the previous section. The rough general idea is to consider signed formulas of the forms F:φ and T:φ, and explore again the full capabilities of the classical metalinguistic notation used in the last section, in the following way:

(P0) Exchange each expression of the form $b(\oplus(\overrightarrow{\varphi})) = V_n$ for a signed formula of the form $\mathrm{V}_n{:}\oplus(\overrightarrow{\varphi})$.

(P1) Treat the translation $\mathsf{tab}[\oplus]\langle\mathrm{V}\rangle$ of a given axiom $\mathsf{biv}[\oplus]\langle V\rangle$ as a tableau rule for $\oplus$: at the left-hand side of each meta-implication you find a signed formula that should be matched to a node of a given branch; the meta-disjuncts on its right-hand side describe the content of distinct branches generated by the application of the rule, each of which will contain a collection of signed formulas.

(P2) Treat the translation $\mathsf{tab}[\mathrm{C}n]$ of a given axiom $\mathsf{biv}[Cn]$ as a closure rule: at the left-hand of each meta-implication you find a collection of signed

† Check `http://tinyurl.com/5cakro`.

formulas which allow you to declare a given branch closed once you can match all the former formulas to nodes of the latter branch.

In the case of G_3, the procedure will produce the following tableau rules.

$$\mathsf{tab}[\to]\langle\mathrm{F}\rangle\ \frac{\mathrm{F}{:}\alpha{\to}\beta}{\mathrm{F}{:}\neg\alpha,\mathrm{T}{:}\neg\beta \mid \mathrm{T}{:}\alpha,\mathrm{F}{:}\beta} \qquad \mathsf{tab}[\neg\to]\langle\mathrm{F}\rangle\ \frac{\mathrm{F}{:}\neg(\alpha{\to}\beta)}{\mathrm{T}{:}\neg\alpha \mid \mathrm{F}{:}\neg\beta}$$

$$\mathsf{tab}[\to]\langle\mathrm{T}\rangle\ \frac{\mathrm{T}{:}\alpha{\to}\beta}{\mathrm{T}{:}\neg\alpha \mid \mathrm{T}{:}\beta \mid \mathrm{F}{:}\alpha,\mathrm{F}{:}\neg\beta} \qquad \mathsf{tab}[\neg\to]\langle\mathrm{T}\rangle\ \frac{\mathrm{T}{:}\neg(\alpha{\to}\beta)}{\mathrm{F}{:}\neg\alpha,\mathrm{T}{:}\neg\beta}$$

$$\mathsf{tab}[\neg\neg]\langle\mathrm{F}\rangle\ \frac{\mathrm{F}{:}\neg(\neg\alpha)}{\mathrm{T}{:}\neg\alpha} \qquad \mathsf{tab}[\mathrm{C}0]\ \frac{\mathrm{F}{:}\alpha,\mathrm{T}{:}\alpha}{\divideontimes}$$

$$\mathsf{tab}[\neg\neg]\langle\mathrm{T}\rangle\ \frac{\mathrm{T}{:}\neg(\neg\alpha)}{\mathrm{F}{:}\neg\alpha} \qquad \mathsf{tab}[\mathrm{C}1]\ \frac{\mathrm{T}{:}\alpha,\mathrm{T}{:}\neg\alpha}{\divideontimes}$$

As expected, we will say that $b \in \mathsf{Sem}_2^{G_3}$ *satisfies* a signed formula $V{:}\varphi$ exactly when $b(\varphi) = V$. Also, we will say that a set R of signed formulas is *satisfiable* if its signed formulas can be jointly satisfied by some fixed bivaluation. Recall that a branch in a tableau is said to be *exhausted* if it is closed and a tableau rule has been applied to every formula of positive depth.

Adequation. The following properties hold of the tableau system for G_3, given a root set R of signed formulas.

> **Soundness.** If all the branches in a tableau with root R are closed then R is not satisfiable.
>
> **Completeness.** If an exhausted branch in a tableau with root R is not closed then R is satisfiable.
>
> **Termination.** If R is finite then a tableau with root R can be built such that every branch is either exhausted or closed.

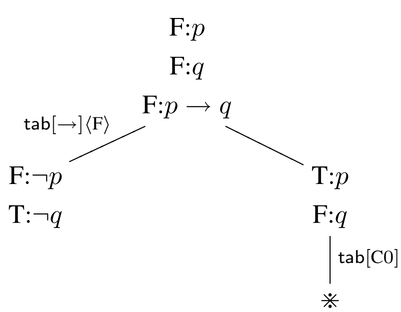
Proof. Soundness is straightforward. Assume, by absurd, that $b \in \mathsf{Sem}_2^{G_3}$ satisfies the root. As by definition b verifies the conditions posed by the clauses from which the tableau rules are built, then, using a simple inductive argument, b must satisfy some of the branches of the tableau. But if all the branches are closed, this configures a non-satisfiable situation for b, according to the closure axioms (Cn). Thus, the root cannot be satisfied.

Completeness is a corollary of the effectiveness of the bivaluation semantics. Consider any exhausted non-closed branch of a tableau with root R.

Take all the depth 0 formulas and consider any $b \in \mathsf{Sem}_2^{G_3}$ satisfying them (such a b must exist as it obviously satisfies all the closure axioms). Note that each bivaluation clause is indeed an equivalence, as the right-hand sides of the F and T cases are disjoint and all other possibilities are excluded by the closure axioms. Thus, if b satisfies one of the concluding branches of a rule it also satisfies the premise. As all the possible rules have been applied, the result follows easily by induction on the construction of the tableau.

The procedure for termination is very simple. Just apply all possible rules in branches that are not closed. Since every formula of positive depth has exactly one applicable rule, in the present example of G_3, and each rule application produces branches with formulas of lesser depth, the process is clearly bound to terminate. □

We shall use the setting posed by the tableau system for G_3 to show how the classic-like 2-signed framework we adopted enables one to compare logics. One possibility would be to confront G_3 with Łukasiewicz's logic $Ł_3$, by showing that $\mathrm{F}{:}p, \mathrm{F}{:}\neg p, \mathrm{T}{:}\neg q \models_{Ł_3} \mathrm{F}{:}\neg(p \to q)$ and $\mathrm{F}{:}p, \mathrm{F}{:}\neg p, \mathrm{T}{:}\neg q \models_{G_3} \mathrm{T}{:}\neg(p \to q)$, but we would first have to develop the example of $Ł_3$ (this has been done in detail in [16] and [36]). We could also distinguish G_3 from classical logic by showing $\not\models_{G_3} \mathrm{T}{:}((p \to q) \to p) \to p$. We shall, however, illustrate here the functioning of our framework by pinpointing the exact way in which the interpretation of G_3 (or intuitionistic) implication departs from the truth-table of classical implication and using the above tableau system for G_3 to show that $\mathrm{F}{:}p, \mathrm{F}{:}q \not\models_{G_3} \mathrm{T}{:}p \to q$.



The open branch on the left allows us to extract the unique 3-valued countermodel, $w(p) = v_1$ and $w(q) = f$.

A note should be added here on the effect that the separating formulas might have on the termination of the tableau systems generated by our axiom extraction procedures. As remarked in [16], in all cases in which such separating formulas are introduced *by abbreviation*, with the help of the primitive constructors of the language of a given logic, the systems obtained will allow for some non-determinism in the choice of rules that build derivations. The reason is that in such a case there will typically be circumstances in which the heads of more than one rule match the same node. In that case, a bad choices of rules to be applied by the user in constructing his derivation could in fact result in non-termination. To deal with that issue, and to guarantee deduction in the new systems to be completely automatic, our second extraction algorithm was in fact associated to a convenient *proof strategy* based, in each case, on a non-canonical complexity / depth measure of the formulas involved. The aim is to make the corresponding tableau systems 'analytic', in an extended sense of the term, once the adherence to the mentioned proof strategies *does* guarantee termination of the task of verifying the validity of a given inference. Moreover, as usual, when any given such terminated task produces a non-closed tableau, exhausted according to the new definition of complexity measure, full counter-models may be promptly extracted from the open branches of the tableau, as exemplified above. The challenges raised by the implementation of such a proof strategy were taken up-front in [34].

Finally, going back one last time to a topic discussed in the previous sections, a case of special interest, from the viewpoint of our above mentioned axiom extraction algorithms, is the one in which conservative extensions are needed in order to distinguish between algebraic truth-values. Consider, for instance, the $\{\neg, \to\}$-fragment of the Gödel logic G_4. In G_4 we have $\mathcal{V} = \{v_0, v_1, v_2, v_3\}$ and $\mathcal{V}^1 = \{v_3\}$, and abbreviate v_0 by f and v_3 by t. The operators are then defined exactly as in G_3 (section 2). In order to separate the values we need, for example, to *add* a constant $\boxtimes$ such that $\lceil \boxtimes \rceil = v_1$. In the extended language, we can see that the tuples formed from $\langle \lceil p_0 \rceil, \lceil \neg p_0 \rceil, \lceil p_0 \to \boxtimes \rceil \rangle$ map the truth-values v_0, v_1, v_2, v_3 to the binary prints$\langle F, T, T \rangle, \langle F, F, T \rangle, \langle F, F, F \rangle, \langle T, F, T \rangle$, respectively. It is straightforward to produce the bivaluation axioms $\mathsf{biv}[\to]\langle V \rangle$, $\mathsf{biv}[\neg\to]\langle V \rangle$, $\mathsf{biv}[(\to)\to\boxtimes]\langle V \rangle$, $\mathsf{biv}[\neg\neg]\langle V \rangle$, $\mathsf{biv}[(\neg)\to\boxtimes]\langle V \rangle$, for $V \in \{F, T\}$, as well as the closure conditions (this time taking $\boxtimes$ also into account), and the corresponding tableau rules. The complexity measure is defined straightforwardly, by starting with $\mathsf{dpth}(p) = \mathsf{dpth}(\neg p) = \mathsf{dpth}(p \to \boxtimes) = \mathsf{dpth}(\boxtimes) = \mathsf{dpth}(\neg\boxtimes) = \mathsf{dpth}(\boxtimes \to \boxtimes) = 0$. However, given the non-determinism generated by the overlap of the ranges of applicability of the rules $\mathtt{tab}[\to]\langle \mathsf{V} \rangle$ with the rules $\mathtt{tab}[(\to)\to\boxtimes]\langle \mathsf{V} \rangle$ and $\mathtt{tab}[(\neg)\to\boxtimes]\langle \mathsf{V} \rangle$, the depth function which guarantees the analyticity of the tableau construction procedure must give priority to the latter, thus guaranteeing that the former rule is applied only when none of the latter can. The corresponding proof strategy that guarantees termination of the proof system is defined in accordance with such priority. This specific issue is discussed and illustrated in [16, 18].

4 CONCLUDING REMARKS

Analyticity has also been the focus of a recent study on non-deterministic semantics (cf. [1], and recall Section 2), and this study has been showing some interesting counterparts in the modularity of the approach, though still not as much reflected in the development of *uniform* classic-like proof-theoretical frameworks. Canonical multi-signed sequent-style proof systems have been developed, at any rate, for this kind of semantics (cf. [4]). Cut-free classic-like sequent systems adequate for logics presented by way of a bivalent semantics have also been studied elsewhere (cf. [8]). In [12] we are to show how our novel classic-like automated axiomatization procedure may indeed be extended from bivalent semantics extracted from finite-valued logics to all other logics whose semantics can be specified in dyadic format, coupling the obtained proof system, in each case, with a convenient proof strategy originated from a convenient non-canonical complexity measure.

Developments toward the implementation of our updated axiom-extraction procedure, together with a fully automated proof tactic in a computer-assisted environment, are reported in [34]. In [18] we show how such aim may be attained, introducing only 'minimal' changes to the original logic.

Further extensions of such constructive procedures and strategies should target also genuinely infinite-valued logics, logics endowed with other kinds of semantics that generalize the traditional notion of truth-functionality, and first-order logics. Improvements on efficiency of the associated proof systems should be expected if the format of the extracted rules is modified, for instance, in order to have them be produced as KE-tableaux (cf. [24]), allowing for a finer negotiation with the notion of analyticity.

Suszko's Thesis is certainly unavailing if we regard it as a dogma, but it can be an insightful tool of logical analysis, as we hope to have illustrated here. Truth-functionality is for sure a nice and simple rule for our algebraic-oriented minds, but there is no reason to fear its absence, even from a strictly algebraic point of view, as results from recent developments in algebraic logic (cf. [15] and [10]) have shown.

ACKNOWLEDGMENT

The first author was partly supported by FCT and EU FEDER, namely via the KLog project PTDC/MAT/ 68723/2006 of SQIG-IT. The second author acknowledges partial support by CAPES and CNPq. A preliminary version of the present survey was presented at [17].

REFERENCES

[1] Arnon Avron. (2009). Multi-valued semantics: Why and how. *Studia Logica*, 92:163–182.

[2] Arnon Avron, Jonathan Ben-Naim, and Beata Konikowska. (2007). Cut-free ordinary sequent calculi for logics having generalized finite-valued semantics. *Logica Universalis*, 1:41–70.

[3] Arnon Avron and Iddo Lev. (2004). Non-deterministic matrices. In *Proceedings of the* XXXIV *International Symposium on Multiple-Valued Logic* (ISMVL 2004), pages 282–287. IEEE Computer Society.

[4] Arnon Avron and Anna Zamansky. (to appear). Non-deterministic semantics for logical systems (a survey). In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*. Kluwer, 2nd edition.

[5] Matthias Baaz, Christian G. Fermüller, and Gernot Salzer. (2001). Automated deduction for many-valued logics. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 1355–1402. Elsevier and MIT Press.

[6] Diderik Batens. (1982). A bridge between two-valued and many-valued semantic systems: n-tuple semantics. In *Proceedings of the* XII *International Symposium on Multiple-Valued Logic* (ISMVL 1982), pages 318–322. IEEE Computer Society.

[7] Jean-Yves Béziau. (1998). Recherches sur la logique abstraite: Les logiques normales. *Logika*, 18:105–114.

[8] Jean-Yves Béziau. (2001). Sequents and bivaluations. *Logique et Analyse (N.S.)*, 44(176):373–394.

[9] W. Blok and D. Pigozzi. (1989). Algebraizable logics. *Memoirs of the AMS*, 77(396).

[10] C. Caleiro and R. Gonçalves. Towards a behavioral algebraic theory of logical valuations. In print.

[11] Carlos Caleiro, Walter Carnielli, Marcelo E. Coniglio, and João Marcos. (2005). Two's company: "The humbug of many logical values". In J.-Y. Béziau, editor, *Logica Universalis*, pages 169–189. Birkhäuser Verlag, Basel, Switzerland. Preprint available at: `http://sqig.math.ist.utl.pt/pub/CaleiroC/05-CCCM-dyadic.pdf`.

[12] Carlos Caleiro, Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. How many logical values are there? Dyadic semantics for many-valued logics. Draft. Forthcoming.

[13] Carlos Caleiro, Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. (2003). Dyadic semantics for many-valued logics. Research report, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT. Presented at the III World Congress on Paraconsistency, Toulouse, FR, July 28–31, 2003. `http://sqig.math.ist.utl.pt/pub/CaleiroC/03-CCCM-dyadic2.pdf`.

[14] Carlos Caleiro, Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. (2003). Suszko's Thesis and dyadic semantics. Research report, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT. Presented at the III World Congress on Paraconsistency, Toulouse, FR, July 28–31, 2003. `http://sqig.math.ist.utl.pt/pub/CaleiroC/03-CCCM-dyadic1.pdf`.

[15] Carlos Caleiro and Ricardo Gonçalves. (2009). Algebraic valuations as behavioral logical matrices. In H. Ono, M. Kanazawa, and R. de Queiroz, editors, *Proceedings of the* XVI *Workshop on Logic, Language, Information and Computation* (WoLLIC 2009), held in Tokyo, JP, June 2009, volume 5514 of *Lecture Notes in Artificial Intelligence*, pages 13–25. Springer. Preprint available at:
http://sqig.math.ist.utl.pt/pub/CaleiroC/09-CG-AlgValuations.pdf.

[16] Carlos Caleiro and João Marcos. (2009). Classic-like analytic tableaux for finite-valued logics. In H. Ono, M. Kanazawa, and R. de Queiroz, editors, *Proceedings of the* XVI *Workshop on Logic, Language, Information and Computation* (WoLLIC 2009), held in Tokyo, JP, June 2009, volume 5514 of *Lecture Notes in Artificial Intelligence*, pages 268–280. Springer. Preprint available at:
http://sqig.math.ist.utl.pt/pub/CaleiroC/09-CM-ClATab4FVL.pdf.

[17] Carlos Caleiro and João Marcos. (2010). Two many values: An algorithmic outlook on Suszko's Thesis. In *Proceedings of the* XL *International Symposium on Multiple-Valued Logic* (ISMVL 2010), pages 93–97. IEEE Computer Society.

[18] Carlos Caleiro and João Marcos, (2010). A uniform classic-like analytic deductive formalism for finite-valued logics. Submitted to publication.

[19] Walter Carnielli. (1990). Many-valued logics and plausible reasoning. In *Proceedings of the* XX *International Symposium on Multiple-Valued Logic,* held at the University of Charlotte / NC, US, 1990, pages 328–335. IEEE Computer Society.

[20] Walter A. Carnielli. (1987). Systematization of the finite many-valued logics through the method of tableaux. *The Journal of Symbolic Logic*, 52(2):473–493.

[21] Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. (2007). Logics of Formal Inconsistency. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 14, pages 1–93. Kluwer, 2nd edition. Preprint available at:
http://sqig.math.ist.utl.pt/pub/MarcosJ/03-CCM-lfi.pdf.

[22] Newton C. A. da Costa and Elias Alves. (1977). A semantical analysis of the calculi C_n. *Notre Dame Journal of Formal Logic*, 18(4):621–630.

[23] Newton C. A. da Costa, Jean-Yves Béziau, and Otávio A. S. Bueno. (1996). Malinowski and Suszko on many-valued logics: on the reduction of many-valuedness to two-valuedness. *Modern Logic*, 3:272–299.

[24] Marcello D'Agostino and Marco Mondadori. (1994). The taming of the cut: classical refutations with analytic cut. *Journal of Logic and Computation*, 4(3):285–319.

[25] Reiner Hähnle. (1999). Tableaux for many-valued logics. In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 529–580. Springer.

[26] Andrea Loparić and Newton C. A. da Costa. (1984). Paraconsistency, paracompleteness, and valuations. *Logique et Analyse (N.S.)*, 27(106):119–131.

[27] Jerzy Łoś and Roman Suszko. (1958). Remarks on sentential logics. *Indagationes Mathematicae*, 20:177–183.

[28] G. Malinowski. (1977). Classical characterization of n-valued Łukasiewicz calculi. *Reports on Mathematical Logic*, 9:41–45.

[29] Grzegorz Malinowski. (1993). *Many-Valued Logics*, volume 25 of *Oxford Logic Guides*. Clarendon Press, Oxford.

[30] Grzegorz Malinowski. (1994). Inferential many-valuedness. In J. Woleński, editor, *Philosophical Logic in Poland*, pages 75–84. Kluwer, Dordrecht.

[31] João Marcos. (2004). Possible-translations semantics. In W. A. Carnielli, F. M. Dionísio, and P. Mateus, editors, *Proceedings of the Workshop on Combination of Logics: Theory and applications* (CombLog'04), held in Lisbon, PT, July 28–30, 2004, pages 119–128. Departamento de Matemática, Instituto Superior Técnico. Extended version available at: `http://sqig.math.ist.utl.pt/pub/MarcosJ/04-M-pts.pdf`.

[32] João Marcos. (2008). Possible-translations semantics for some weak classically-based paraconsistent logics. *Journal of Applied Non-Classical Logics*, 18(1):7–28. Preprint available at: `http://sqig.math.ist.utl.pt/pub/MarcosJ/04-M-PTS4swcbPL.pdf`.

[33] João Marcos. (2009). What is a non-truth-functional logic? *Studia Logica*, 92:215–240.

[34] João Marcos. (2010). Automatic generation of proof tactics for finite-valued logics. *Electronic Proceedings in Theoretical Computer Science*, 21:91–98. Available at: `http://arxiv.org/abs/1003.4802v1`.

[35] João Marcos. (2010). The value of the two values. Extended abstract, LoLITA and DIMAp / UFRN, and Theory and Logic Group, TU-Wien. Presented at 31st Linz Seminar on Fuzzy Set Theory: Lattice-valued logic and its applications (LINZ 2010), Feb 9–13, 2009, Linz, AT. Submitted for publication.

[36] João Marcos and Dalmo Mendonça. (2009). Towards fully automated axiom extraction for finite-valued logics. In W. Carnielli, M. E. Coniglio, and I. M. L. D'Ottaviano, editors, *The Many Sides of Logic*, Studies in Logic, pages 425–440. College Publications, London. Preprint available at: `http://sqig.math.ist.utl.pt/pub/MarcosJ/08-MM-towards.pdf`.

[37] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. (2002). *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer.

[38] D. J. Shoesmith and Timothy J. Smiley. (1971). Deducibility and many-valuedness. *The Journal of Symbolic Logic*, 36(4):610–622.

[39] Stanisław J. Surma. (1974). An algorithm for axiomatizing every finite logic. In David C. Rine, editor, *Computer Science and Multiple-Valued Logics,* Selected Papers from the IV *International Symposium on Multiple-Valued Logics (ISMVL 1974)*, pages 143–149. North-Holland, Amsterdam. 2nd edition, 1984.

[40] Roman Suszko. (1972). Abolition of the Fregean Axiom. In R. Parikh, editor, *Logic Colloquium: Symposium on Logic held at Boston,* 1972–73, volume 453 of *Lecture Notes in Mathematics*, pages 169–239. Springer-Verlag, Berlin.

[41] Roman Suszko. (1975). Remarks on Łukasiewicz's three-valued logic. *Bulletin of the Section of Logic*, 4:87–90.

[42] Roman Suszko. (1977). The Fregean Axiom and Polish mathematical logic in the 1920s. *Studia Logica*, 36:377–380.

[43] Heinrich Wansing and Yaroslav Shramko. (2008). Suszko's Thesis, inferential many-valuedness, and the notion of a logical system. *Studia Logica*, 88(3):405–429.

[44] Ryszard Wójcicki. (1988). *Theory of Logical Calculi*. Kluwer, Dordrecht.