J Philos Logic (2015) 44:473–487
DOI 10.1007/s10992-014-9337-3

# Analytic Tableaux for all of *SIXTEEN*$_3$

**Reinhard Muskens · Stefan Wintein**

Received: 19 July 2014 / Accepted: 10 November 2014 / Published online: 4 December 2014

**Abstract** In this paper we give an analytic tableau calculus $\mathbf{PL_{16}}$ for a functionally complete extension of Shramko and Wansing's logic. The calculus is based on signed formulas and a single set of tableau rules is involved in axiomatising each of the four entailment relations $\models_t$, $\models_f$, $\models_i$, and $\models$ under consideration—the differences only residing in initial assignments of signs to formulas. Proving that two sets of formulas are in one of the first three entailment relations will in general require developing four tableaux, while proving that they are in the $\models$ relation may require six.

**Keywords** Trilattice *SIXTEEN*$_3$ · Tableau calculi · Functional completeness · Truth entailment · Falsity entailment · Information entailment

## 1 Introduction

In [14] Yaroslav Shramko and Heinrich Wansing define an attractive generalisation of Belnap's well-known four-valued logic (see [3, 4]). Where Belnap generalises classical two-valued propositional logic and moves from the set of truth values $\mathbf{2} = \{1, 0\}$ to the set $\mathbf{4} = \mathscr{P}(\mathbf{2})$ of combinations of truth values, Shramko & Wansing repeat the move and consider logics based on $\mathbf{16} = \mathscr{P}(\mathbf{4})$—combinations of combinations of classical truth values. Belnap's original motivation for the move from $\mathbf{2}$ to $\mathbf{4}$—a logic

R. Muskens (✉)
Tilburg Center for Logic, Ethics, and Philosophy of Science (TiLPS), Tilburg University, Tilburg, Netherlands
e-mail: r.a.muskens@uvt.nl

S. Wintein
Faculty of Philosophy, Erasmus University Rotterdam, Rotterdam, Netherlands
e-mail: stefanwintein@gmail.com

based on this set can model how a computer should deal with the incomplete and/or inconsistent information that may be fed to it—is extended as well, as the logic based on **16** arguably models how a network should reason on the basis of input it gets from computers reasoning on the basis of **4**.

These moves from **2** to **4** and from **4** to **16** come with an increasing number of lattice orderings. There is of course the usual ordering on **2**, but Belnap considers two orders on **4**. The first of these is that of the *logical* lattice L4, in which the values $\mathbf{T} = \{1\}$, $\mathbf{F} = \{0\}$, $\mathbf{N} = \varnothing$, and $\mathbf{B} = \{1, 0\}$ are ordered in a way that corresponds to an extension of the Strong Kleene evaluation scheme (see Fig. 1 for a Hasse diagram). The second is the ordering of the *approximation of information* lattice A4, in which the same values are ordered according to their degree of information. Together these two orderings form what is called a *bilattice*. On **16** Shramko & Wansing consider *three* orderings, denoted with $\leq_t$, $\leq_f$, and $\leq_i$, together forming the so-called *trilattice SIXTEEN*$_3$. Of these three, $\leq_t$ can be interpreted as a *truth* order, $\leq_f$ as a *falsity* order, and $\leq_i$ again as an *approximation of information* order. The lattice L4, which in fact acts as a *truth and falsity* order, is thus replaced by two independent orderings, not each other's inverses, as will become apparent below.

Meet and join operations on these lattices can naturally be associated with conjunction and disjunction operators, but there are unary functions that can interpret negation too. Belnap [4] already considers a negation operation on **4** that swaps **T** and **F**, but leaves **N** and **B** untouched, but it has a natural dual in the operation that swaps **N** and **B** but leaves **T** and **F** as is. On $SIXTEEN_3$ there are three involutions $-_t$, $-_f$, and $-_i$ that are natural candidates to provide the semantics for negation connectives in a logical language.

We thus see that $SIXTEEN_3$ is well-understood (more precise definitions are to follow after this introduction). But the proof theory of logics based on this trilattice is another matter (see also Odintsov & Wansing's [10] section 1.2 on 'the axiomatization problem'). We are aware of several approaches. Odintsov [9], for example, gives Hilbert style axiomatisations of the truth entailment relation $\models_t$ naturally based on $\leq_t$ and the falsity entailment relation $\models_f$ based on $\leq_f$. Another way of dealing with this problem is exemplified in Wansing [16], where cut-free sequent calculi for $\models_t$ and $\models_f$ are presented. And a third way can be found in Odintsov & Wansing's [10], where a so-called 'bi-calculus' for the $t$ and $f$ consequence relations is given. More

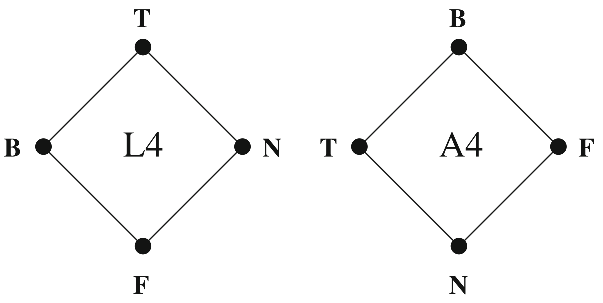

**Fig. 1** Belnap's lattices L4 and A4

sequent and tableau systems for $SIXTEEN_3$ related logics are presented in Kamide & Wansing [5] and in Wansing & Kamide [18].

But something is still to be wished for, as the logical languages that have been considered do not provide a set of connectives that is functionally complete for **16**. In particular, we are not aware of an existing system in which a connective $\wedge_i$ corresponding to meet in the $\leq_i$ order is definable. (Odintsov's and Wansing's logics are based on connectives corresponding to meet and join in the $\leq_t$ and $\leq_f$ orderings, plus, in the case of some logics, certain implications $\rightarrow_t$ and $\rightarrow_f$ that are residua with respect to these orderings. But $\wedge_i$ cannot be defined in any of the resulting languages, as will become clear below). Since $\wedge_i$ can be thought of as formalising a sceptical information merging strategy that one may reasonably want a computer network to follow, there is some interest in clarifying its proof theory and characterising its interactions with other connectives.

The purpose of this paper therefore is to give a calculus that characterises a propositional logic that expresses all of $SIXTEEN_3$, i.e. deals with all truth functions on **16**. We will provide such a logic with the help of a generalisation of the approach in Wintein & Muskens [19], where a calculus for a functionally complete first-order logic based on Belnap's two lattices was given. That paper was based on two ideas that are relevant here. The first of these—taken over from Muskens [8] and ultimately from Langholm [6]—was to use a four-sided sequent calculus,[1] the second idea was to let each proof correspond to *two* proof trees in the usual sense (see also Wintein & Muskens [20]). We will generalise this approach by giving a signed analytic tableau calculus based on *eight* signs and by letting proofs correspond to *four* or even *six* tableaux (an eight-sided Gentzen sequent variant could easily be given). From a semantic point of view, each of the eight signs (for which we use $\mathsf{t}$, $\mathsf{f}$, $\mathsf{n}$, $\mathsf{b}$, $\not\mathsf{t}$, $\not\mathsf{f}$, $\not\mathsf{n}$, and $\not\mathsf{b}$) signals whether an element from $\mathbf{4} = \{\mathbf{T}, \mathbf{F}, \mathbf{N}, \mathbf{B}\}$ is present or absent in the value of the sentence that is signed and each tableau checks whether one of these four values is transmitted from premises to conclusions or vice versa. While the description just given may seem to point to a rather complicated system, the calculus is in fact simple and can be used to provide syntactic characterisations of the semantic entailment relations $\models_t$, $\models_f$, and $\models_i$ (information entailment[2]) based on the three lattice orderings, and a relation $\models$ which is the intersection of $\models_t$ and $\models_f$. A single set of rules can be used for all four consequence relations (a feature our logic shares with the sequent calculus framework of Wansing [16]), but testing for different consequence relations involves different ways of initially assigning signs to formulas.

The presentation in this paper is meant to be technically self-contained, but is not self-contained in terms of providing general background and motivation. For

[1]Many-sided calculi for many-valued logics also result from the methods used in, for example, [1, 12, 13, 21], but [8] explains why these methods are not applicable in the case of a functionally complete variant of Belnap's logic. Wansing [16] also makes use of many-sided sequents.

[2]We use the terms 'entailment relation' and 'consequence relation' purely for convenience in the case of the information entailment relation $\models_i$. Strictly more correct perhaps would be 'necessary approximation', a term that was used in [19] for a 'consequence' relation based on Belnap's A4 lattice.

these, the reader is invited to consult the papers cited above, or the somewhat more philosophical Wansing [17], or the full monograph Shramko & Wansing [15].

## 2 The Trilattice $SIXTEEN_3$

The general theory of trilattices is subtle, but a description of $SIXTEEN_3$ in particular is easy to give. As explained in the introduction, the domain of this trilattice is $\mathbf{16} = \mathscr{P}(\mathbf{4})$, where $\mathbf{4} = \mathscr{P}(\{1, 0\}) = \{\mathbf{T}, \mathbf{F}, \mathbf{N}, \mathbf{B}\}$. In order to obtain the three lattice orderings, the following auxiliary definition from [14] is useful.

**Definition 1** For any $S \in \mathbf{16}$, let

$$\begin{aligned} S^t &:= \{x \in S \mid 1 \in x\} \quad (= S \cap \{\mathbf{T}, \mathbf{B}\}) \\ S^{-t} &:= \{x \in S \mid 1 \notin x\} \quad (= S \cap \{\mathbf{F}, \mathbf{N}\}) \\ S^f &:= \{x \in S \mid 0 \in x\} \quad (= S \cap \{\mathbf{F}, \mathbf{B}\}) \\ S^{-f} &:= \{x \in S \mid 0 \notin x\} \quad (= S \cap \{\mathbf{T}, \mathbf{N}\}) \end{aligned}$$

The idea here is that $S^t$ consists of all values in $S$ that code for truth, $S^{-t}$ for all values in $S$ that do not, $S^f$ for all values that code for falsity, and $S^{-f}$ for those that do not code for falsity. The trilattice orderings are now obtained as follows with the help of these operations.

**Definition 2** Define $\leq_t$, $\leq_f$, and $\leq_i$ by letting, for any $S_1, S_2 \in \mathbf{16}$:

$$\begin{aligned} S_1 \leq_t S_2 &\iff S_1^t \subseteq S_2^t \text{ and } S_2^{-t} \subseteq S_1^{-t} \\ S_1 \leq_f S_2 &\iff S_2^f \subseteq S_1^f \text{ and } S_1^{-f} \subseteq S_2^{-f} \\ S_1 \leq_i S_2 &\iff S_1 \subseteq S_2 \end{aligned}$$

Note that $\leq_t$ is defined in terms of truth only, $\leq_f$ in terms of falsity only. We have followed Odintsov [9] and Odintsov & Wansing [10] in letting $\leq_f$ be the inverse of the $\leq_f$ relation originally defined in [14]. An increase in the falsity order will make statements *less* false, not more so (and the falsity order could equally well be called a *nonfalsity* order).

For our purposes it will be worthwile to also characterise $\leq_t$ and $\leq_f$ in the following more concrete way. That these equivalences hold is verified by an easy inspection.

**Proposition 1** *For any $S_1$, $S_2$ in $SIXTEEN_3$:*

$$S_1 \leq_t S_2 \iff \begin{cases} \mathbf{T} \in S_1 \Rightarrow \mathbf{T} \in S_2 \ \textit{and} \ \mathbf{B} \in S_1 \Rightarrow \mathbf{B} \in S_2 \ \textit{and} \\ \mathbf{F} \in S_2 \Rightarrow \mathbf{F} \in S_1 \ \textit{and} \ \mathbf{N} \in S_2 \Rightarrow \mathbf{N} \in S_1 \end{cases}$$

$$S_1 \leq_f S_2 \iff \begin{cases} \mathbf{F} \in S_2 \Rightarrow \mathbf{F} \in S_1 \ \textit{and} \ \mathbf{B} \in S_2 \Rightarrow \mathbf{B} \in S_1 \ \textit{and} \\ \mathbf{T} \in S_1 \Rightarrow \mathbf{T} \in S_2 \ \textit{and} \ \mathbf{N} \in S_1 \Rightarrow \mathbf{N} \in S_2 \end{cases}$$

In fact the four implications characterising $\leq_t$ will lead to four auxiliary semantic entailment relations in Section 4, which will, in their turn, correspond to four syntactic consequence relations.

Each of the lattice orderings on **16** comes with meet and join operations, We will write $\sqcap_x$ for the meet and $\sqcup_x$ for the join of $\leq_x$ $(x \in \{t, f, i\})$. Clearly, $\sqcap_i$ is $\cap$ and $\sqcup_i$ is $\cup$. For the other four operations the following proposition provides a handy reference (see also [15], but note that the inverse of our $\leq_f$ is employed there).

**Proposition 2** *For any $S$, $S'$ in $SIXTEEN_3$:*

$$\begin{array}{llll}
\mathbf{T} \in S \sqcap_t S' \Leftrightarrow \mathbf{T} \in S \textit{ and } \mathbf{T} \in S'; & & \mathbf{T} \in S \sqcup_t S' \Leftrightarrow \mathbf{T} \in S \textit{ or } \mathbf{T} \in S'; \\
\mathbf{B} \in S \sqcap_t S' \Leftrightarrow \mathbf{B} \in S \textit{ and } \mathbf{B} \in S'; & & \mathbf{B} \in S \sqcup_t S' \Leftrightarrow \mathbf{B} \in S \textit{ or } \mathbf{B} \in S'; \\
\mathbf{F} \in S \sqcap_t S' \Leftrightarrow \mathbf{F} \in S \textit{ or } \mathbf{F} \in S'; & & \mathbf{F} \in S \sqcup_t S' \Leftrightarrow \mathbf{F} \in S \textit{ and } \mathbf{F} \in S'; \\
\mathbf{N} \in S \sqcap_t S' \Leftrightarrow \mathbf{N} \in S \textit{ or } \mathbf{N} \in S'; & & \mathbf{N} \in S \sqcup_t S' \Leftrightarrow \mathbf{N} \in S \textit{ and } \mathbf{N} \in S';
\end{array}$$

$$\begin{array}{llll}
\mathbf{T} \in S \sqcap_f S' \Leftrightarrow \mathbf{T} \in S \textit{ and } \mathbf{T} \in S'; & & \mathbf{T} \in S \sqcup_f S' \Leftrightarrow \mathbf{T} \in S \textit{ or } \mathbf{T} \in S'; \\
\mathbf{B} \in S \sqcap_f S' \Leftrightarrow \mathbf{B} \in S \textit{ or } \mathbf{B} \in S'; & & \mathbf{B} \in S \sqcup_f S' \Leftrightarrow \mathbf{B} \in S \textit{ and } \mathbf{B} \in S'; \\
\mathbf{F} \in S \sqcap_f S' \Leftrightarrow \mathbf{F} \in S \textit{ or } \mathbf{F} \in S'; & & \mathbf{F} \in S \sqcup_f S' \Leftrightarrow \mathbf{F} \in S \textit{ and } \mathbf{F} \in S'; \\
\mathbf{N} \in S \sqcap_f S' \Leftrightarrow \mathbf{N} \in S \textit{ and } \mathbf{N} \in S'; & & \mathbf{N} \in S \sqcup_f S' \Leftrightarrow \mathbf{N} \in S \textit{ or } \mathbf{N} \in S'.
\end{array}$$

$SIXTEEN_3$ can now be defined as $\langle \mathbf{16}, \sqcap_t, \sqcup_t, \sqcap_f, \sqcup_f, \sqcap_i, \sqcup_i \rangle$ and the reader may wish to verify that, for any $\bullet, \circ \in \{\sqcap_t, \sqcup_t, \sqcap_f, \sqcup_f, \sqcap_i, \sqcup_i\}$, we have that $a \circ (b \bullet c) = (a \circ b) \bullet (a \circ c)$ (see also Rivieccio [11]). This makes $SIXTEEN_3$ into a *distributive* trilattice, from which it can be shown to follow that the structure is *interlaced*, in the sense that all six lattice operations are monotone with respect to the three lattice orders.

It is useful to also have available three unary operations that can be taken to be the semantic correlates of negation connectives. The following definition provides them.

**Definition 3** Define the unary operations $-_t$, $-_f$, and $-_i$ by letting, for any $S \in \mathbf{16}$:

$$\begin{array}{lll}
\mathbf{T} \in -_t S \Leftrightarrow \mathbf{N} \in S; & \mathbf{T} \in -_f S \Leftrightarrow \mathbf{B} \in S; & \mathbf{T} \in -_i S \Leftrightarrow \mathbf{F} \notin S; \\
\mathbf{B} \in -_t S \Leftrightarrow \mathbf{F} \in S; & \mathbf{B} \in -_f S \Leftrightarrow \mathbf{T} \in S; & \mathbf{B} \in -_i S \Leftrightarrow \mathbf{N} \notin S; \\
\mathbf{F} \in -_t S \Leftrightarrow \mathbf{B} \in S; & \mathbf{F} \in -_f S \Leftrightarrow \mathbf{N} \in S; & \mathbf{F} \in -_i S \Leftrightarrow \mathbf{T} \notin S; \\
\mathbf{N} \in -_t S \Leftrightarrow \mathbf{T} \in S; & \mathbf{N} \in -_f S \Leftrightarrow \mathbf{F} \in S; & \mathbf{N} \in -_i S \Leftrightarrow \mathbf{B} \notin S;
\end{array}$$

Inspection will show that, for each pairwise distinct $x, y \in \{t, f, i\}$,

$$\begin{aligned}
&\text{if } a \leq_x b, \text{ then } -_x b \leq_x -_x a\,; \\
&\text{if } a \leq_y b, \text{ then } -_x a \leq_y -_x b\,; \\
&a = -_x -_x a.
\end{aligned}$$

(Again see Rivieccio [11].)

This ends the description of $SIXTEEN_3$ proper, but before we end this section we would like to introduce three operators defined by Odintsov in [9], as they will play a role in the next section. The first two can be defined as follows.

**Definition 4** Define the binary operations $\sqsupset_t$, and $\sqsupset_f$ by letting, for any $S, S' \in \mathbf{16}$:

$$\begin{array}{llll}
\mathbf{T} \in S \sqsupset_t S' \Leftrightarrow \mathbf{T} \notin S \text{ or } \mathbf{T} \in S'; & & \mathbf{T} \in S \sqsupset_f S' \Leftrightarrow \mathbf{T} \notin S \text{ or } \mathbf{T} \in S'; \\
\mathbf{B} \in S \sqsupset_t S' \Leftrightarrow \mathbf{B} \notin S \text{ or } \mathbf{B} \in S'; & & \mathbf{B} \in S \sqsupset_f S' \Leftrightarrow \mathbf{B} \notin S \text{ and } \mathbf{B} \in S'; \\
\mathbf{F} \in S \sqsupset_t S' \Leftrightarrow \mathbf{F} \notin S \text{ and } \mathbf{F} \in S'; & & \mathbf{F} \in S \sqsupset_f S' \Leftrightarrow \mathbf{F} \notin S \text{ and } \mathbf{F} \in S'; \\
\mathbf{N} \in S \sqsupset_t S' \Leftrightarrow \mathbf{N} \notin S \text{ and } \mathbf{N} \in S'; & & \mathbf{N} \in S \sqsupset_f S' \Leftrightarrow \mathbf{N} \notin S \text{ or } \mathbf{N} \in S'.
\end{array}$$

Odintsov's unary operator $\rightharpoondown$, which completes the three, is defined by letting $\rightharpoondown S = \{\mathbf{T}, \mathbf{F}, \mathbf{N}, \mathbf{B}\} - S$, for any $S \in \mathbf{16}$.

The reader may note that $\rightharpoondown$ in fact equals the composition of $-_t$, $-_f$, and $-_i$ (in any order) and that $S \sqsupset_t S' = \rightharpoondown S \sqcup_t S'$, while $S \sqsupset_f S' = \rightharpoondown S \sqcup_f S'$, for any $S, S' \in \mathbf{16}$.

## 3 Syntax, Semantics, and Expressivity

The logical language we shall use in this paper will be $\mathscr{L}_{tfi}$, given by the following BNF form (where $p$ comes from some countably infinite set of propositional constants).

$$\varphi ::= p \mid \sim_t\varphi \mid \sim_f\varphi \mid \sim_i\varphi \mid \varphi \wedge_t \varphi \mid \varphi \wedge_f \varphi \mid \varphi \wedge_i \varphi$$

This set-up may be somewhat spartan, but other connectives can be defined and the set $\{\sim_t, \sim_f, \sim_i, \wedge_t, \wedge_f, \wedge_i\}$ is in fact functionally complete, as we shall see shortly.

Let us provide the language $\mathscr{L}_{tfi}$ with a semantics. The following definition will not come as a surprise.

**Definition 5** A valuation function is a function $V$ from the sentences of $\mathscr{L}_{tfi}$ to $\mathbf{16}$ such that

$$\begin{array}{llll}
V(\varphi \wedge_t \psi) = V(\varphi) \sqcap_t V(\psi); & & V(\sim_t\varphi) = -_t V(\varphi); \\
V(\varphi \wedge_f \psi) = V(\varphi) \sqcap_f V(\psi); & & V(\sim_f\varphi) = -_f V(\varphi); \\
V(\varphi \wedge_i \psi) = V(\varphi) \sqcap_i V(\psi); & & V(\sim_i\varphi) = -_i V(\varphi).
\end{array}$$

Given this interpretation of the language, we clearly can also define connectives denoting the operations $\sqcup_t$, $\sqcup_f$, $\sqcup_i$, $\rightharpoondown$, $\sqsupset_t$, and $\sqsupset_f$ via the obvious De Morgan rules and the equivalences discussed in the previous section.

**Definition 6** The following abbreviations will be in force.

$$\begin{aligned}
\varphi \vee_t \psi &:= \sim_t (\sim_t\varphi \wedge_t \sim_t\psi); \\
\varphi \vee_f \psi &:= \sim_f (\sim_f\varphi \wedge_f \sim_f\psi); \\
\varphi \vee_i \psi &:= \sim_i (\sim_i\varphi \wedge_i \sim_i\psi); \\
\neg\varphi &:= \sim_t\sim_f\sim_i\varphi; \\
\varphi \to_t \psi &:= \neg\varphi \vee_t \psi; \\
\varphi \to_f \psi &:= \neg\varphi \vee_f \psi.
\end{aligned}$$

It is easily verified that $V(\neg\varphi) = \rightharpoondown V(\varphi)$ etc.

It will also be possible to define sentences that constantly denote a given element of **16**, regardless of the valuation. This can be done for each element of **16**. In the following definition we have chosen names for these sentences that serve as a mnemonic for the element denoted (e.g. $V(\mathsf{nfb}) = \{\mathbf{N}, \mathbf{F}, \mathbf{B}\}$, for any $V$).

**Definition 7** Let $p_0$ be some fixed propositional constant. The following abbreviations will be used.

$$\begin{array}{ll}
\mathsf{tb} := p_0 \rightarrow_t p_0; & \mathsf{t} := \mathsf{tb} \vee_f \varnothing; \\
\mathsf{nf} :=\sim_t \mathsf{tb}; & \mathsf{n} := \mathsf{nf} \vee_f \varnothing; \\
\varnothing := \mathsf{tb} \wedge_i \mathsf{nf}; & \mathsf{nt} := \mathsf{tb} \vee_f \mathsf{nf}; \\
\mathsf{nftb} :=\sim_i \varnothing; & \mathsf{ftb} := \mathsf{tb} \wedge_f \mathsf{nftb}; \\
\mathsf{b} := \mathsf{tb} \wedge_f \varnothing; & \mathsf{nfb} := \mathsf{nf} \wedge_f \mathsf{nftb}; \\
\mathsf{ntb} := \mathsf{tb} \vee_f \mathsf{nftb}; & \mathsf{fb} := \mathsf{tb} \wedge_f \mathsf{nf}; \\
\mathsf{f} := \mathsf{nf} \wedge_f \varnothing; & \mathsf{nb} := \mathsf{b} \wedge_t \mathsf{ntb}; \\
\mathsf{nft} := \mathsf{nf} \vee_f \mathsf{nftb}; & \mathsf{ft} := \mathsf{t} \wedge_t \mathsf{ftb};
\end{array}$$

Let us call a sentence $\varphi$ *bivalent* if $V(\varphi) = \{\mathbf{T}, \mathbf{B}\}$ or $V(\varphi) = \{\mathbf{N}, \mathbf{F}\}$ for any valuation $V$ (note that $\{\mathbf{T}, \mathbf{B}\}$ is the top element of the $\leq_t$ order, while $\{\mathbf{N}, \mathbf{F}\}$ is its bottom element). It is useful to have connectives at our disposal that always give a result that is bivalent. For arbitrary $\varphi$, consider the sentence $\mathsf{T}\varphi$ defined as

$$\varphi \wedge_t \sim_i\varphi \wedge_t \sim_f\varphi \wedge_t \sim_f\sim_i\varphi.$$

We find by inspection, using Proposition 2 and Definition 3, that, for any $V$,

$$\begin{aligned}
\mathbf{T} \in V(\mathsf{T}\varphi) &\iff \mathbf{T} \in V(\varphi) \text{ and } \mathbf{F} \notin V(\varphi) \text{ and } \mathbf{B} \in V(\varphi) \text{ and } \mathbf{N} \notin V(\varphi) \\
&\iff \mathbf{B} \in V(\mathsf{T}\varphi) \\
\mathbf{F} \in V(\mathsf{T}\varphi) &\iff \mathbf{F} \in V(\varphi) \text{ or } \mathbf{T} \notin V(\varphi) \text{ or } \mathbf{N} \in V(\varphi) \text{ or } \mathbf{B} \notin V(\varphi) \\
&\iff \mathbf{N} \in V(\mathsf{T}\varphi)
\end{aligned}$$

From this it follows that $V(\mathsf{T}\varphi) = \{\mathbf{T}, \mathbf{B}\}$ iff $V(\varphi) = \{\mathbf{T}, \mathbf{B}\}$ and that $V(\mathsf{T}\varphi) = \{\mathbf{N}, \mathbf{F}\}$ iff $V(\varphi) \neq \{\mathbf{T}, \mathbf{B}\}$. This can be used to define other connectives giving bivalent results.

**Definition 8** We define the following connectives.

$$\begin{aligned}
\varphi \twoheadrightarrow_t \psi &:= \mathsf{T}(\varphi \rightarrow_t \psi) \\
\varphi \equiv \psi &:= (\varphi \twoheadrightarrow_t \psi) \wedge_t (\psi \twoheadrightarrow_t \varphi)
\end{aligned}$$

In [9] Odintsov shows that, for every $S,\ S' \in \mathbf{16},\ S \leq_t S'$ iff $S \sqsupset_t S' = \{\mathbf{T}, \mathbf{B}\}$. From this and the definition of $\varphi \twoheadrightarrow_t \psi$ it follows that, for all $V$, $V(\varphi \twoheadrightarrow_t \psi) = \{\mathbf{T}, \mathbf{B1}\}$ iff $V(\varphi) \leq_t V(\psi)$ and $V(\varphi \twoheadrightarrow_t \psi) = \{\mathbf{N}, \mathbf{F}\}$ iff $V(\varphi) \not\leq_t V(\psi)$. The $\equiv$ connective expresses *identity*, as $V(\varphi \equiv \psi) = \{\mathbf{T}, \mathbf{B}\}$ iff $V(\varphi) = V(\psi)$, and $V(\varphi \equiv \psi) = \{\mathbf{N}, \mathbf{F}\}$ otherwise.

We now turn to the functional completeness theorem, for which we will assume that $p_0, p_1, p_2, \ldots, p_n, \ldots$ is some fixed enumeration of the set of propositional constants ($p_0$ was already used in Definition 7). If $g$ is an n-ary truth function

$g : \mathbf{16}^n \to \mathbf{16}$ and $\varphi$ is an $\mathscr{L}_{tfi}$ formula such that exactly $p_0, p_1, \ldots, p_n$ occur in $\varphi$, we say that $\varphi$ *expresses* $g$ if $V(\varphi) = g(V(p_1), \ldots, V(p_n))$, for each valuation $V$.

**Proposition 3** *Every truth function* $g : \mathbf{16}^n \to \mathbf{16}$ *is expressed by an* $\mathscr{L}_{tfi}$ *formula.*

*Proof* The proof generalises that of Muskens [7] for the four-valued case and proceeds by induction on $n$. Zero-place truth functions can be identified with the elements of **16** and inspection of Definition 7 shows that each element of that set is denoted by a constant defined there. For $S \in \mathbf{16}$ we will use the notation $\overline{S}$ to refer to the term given in Definition 7 that denotes $S$. For the induction step, let $g : \mathbf{16}^{n+1} \to \mathbf{16}$ be an $n + 1$-ary function. For each $S \in \mathbf{16}$ define the n-ary truth function $g_S : \mathbf{16}^n \to \mathbf{16}$ by letting $g_S(S_1, \ldots, S_n) = g(S_1, \ldots, S_n, S)$. Using induction we find that each $g_S$ ($S \in \mathbf{16}$) is expressed by some formula $\varphi_S$. Let $\varphi$ be the formula

$$\bigwedge_{t \atop U \in \mathbf{16}} \sim_t (p_{n+1} \equiv \overline{U}) \vee_t \varphi_U \ ,$$

where $\bigwedge_t$ is used to denote a finite $\wedge_t$-conjunction of formulas. Let $V$ be an arbitrary valuation. Then $V(p_{n+1}) = S$, for some $S \in \mathbf{16}$ and, for any $S' \neq S$, it holds that $V(\sim_t(p_{n+1} \equiv \overline{S'}) \vee_t \varphi_{S'}) = \{\mathbf{T}, \mathbf{B}\}$, while $V(\sim_t(p_{n+1} \equiv \overline{S}) \vee_t \varphi_S) = V(\varphi_S)$. It follows that $V(\varphi) = V(\varphi_S) = g_S(V(p_1), \ldots, V(p_n)) = g(V(p_1), \ldots, V(p_n), S)$, so that $V(\varphi) = g(V(p_1), \ldots, V(p_n), V(p_{n+1}))$, and, since $V$ was arbitrary, that $\varphi$ expresses $g$. □

It is also instructive to see that $\{\sim_t, \sim_f, \sim_i, \wedge_t, \wedge_f, \wedge_i\}$ is a minimal set that is functionally complete.

**Proposition 4** *No proper subset of* $\{\sim_t, \sim_f, \sim_i, \wedge_t, \wedge_f, \wedge_i\}$ *can express every truth function* $g : \mathbf{16}^n \to \mathbf{16}$.

*Proof* In order to see that $\sim_t$ is essential, consider a unary $g$ such that $g(\{\mathbf{T}, \mathbf{B}\}) = \{\mathbf{N}, \mathbf{F}\}$. Each operator underlying one of the connectives $\sim_f$, $\sim_i$, $\wedge_t$, $\wedge_f$, and $\wedge_i$ will return the value $\{\mathbf{T}, \mathbf{B}\}$ when its argument(s) all take that value, so no formula in $\{\sim_f, \sim_i, \wedge_t, \wedge_f, \wedge_i\}$ can express $g$.

That $\wedge_i$ cannot be left out can be seen by considering the set

$$\mathscr{O} := \{\{\mathbf{T}, \mathbf{B}\}, \{\mathbf{N}, \mathbf{F}\}, \{\mathbf{N}, \mathbf{T}\}, \{\mathbf{F}, \mathbf{B}\}\} \ .$$

Each connective in $\{\sim_t, \sim_f, \sim_i, \wedge_t, \wedge_f\}$ will return a value from $\mathscr{O}$ when fed arguments from $\mathscr{O}$. It follows that no formula built up from these connectives can take the value $\varnothing$ when all its propositional constants are assigned elements of $\mathscr{O}$. So no formula in $\{\sim_t, \sim_f, \sim_i, \wedge_t, \wedge_f\}$ can express $\sqcap_i$.

For the other four cases proceed similarly. □

The last proposition of this section shows that some logics considered in the literature (see Odintsov [9] and Wansing's [16]) are equally expressive.

**Proposition 5** *The languages*

$$\begin{array}{rl}
\mathscr{L}_{tf}^{\rightarrow_t}: & \varphi ::= p \mid {\sim_t}\varphi \mid {\sim_f}\varphi \mid \varphi \wedge_t \varphi \mid \varphi \wedge_f \varphi \mid \varphi \rightarrow_t \varphi \\
\mathscr{L}_{tf}^{\rightarrow_f}: & \varphi ::= p \mid {\sim_t}\varphi \mid {\sim_f}\varphi \mid \varphi \wedge_t \varphi \mid \varphi \wedge_f \varphi \mid \varphi \rightarrow_f \varphi \\
\mathscr{L}_{tf}^{*}: & \varphi ::= p \mid {\sim_t}\varphi \mid {\sim_f}\varphi \mid \varphi \wedge_t \varphi \mid \varphi \wedge_f \varphi \mid \varphi \rightarrow_t \varphi \mid \varphi \rightarrow_f \varphi \\
\mathscr{L}_{tf}^{\sim_i}: & \varphi ::= p \mid {\sim_t}\varphi \mid {\sim_f}\varphi \mid {\sim_i}\varphi \mid \varphi \wedge_t \varphi \mid \varphi \wedge_f \varphi
\end{array}$$

*are all equally expressive.*

*Proof* Observe that Definition 6 did not make use of $\wedge_i$ to obtain $\rightarrow_t$ and $\rightarrow_f$, so $\mathscr{L}_{tf}^{\sim_i}$ is at least as expressive as any of $\mathscr{L}_{tf}^{\rightarrow_t}$, $\mathscr{L}_{tf}^{\rightarrow_f}$, or $\mathscr{L}_{tf}^{*}$. To see that, conversely, any of $\mathscr{L}_{tf}^{\rightarrow_t}$, $\mathscr{L}_{tf}^{\rightarrow_f}$, and $\mathscr{L}_{tf}^{*}$ are at least as expressive as $\mathscr{L}_{tf}^{\sim_i}$, use Odintsov's [9] sand Wansing's [16] observations that $\neg\varphi$ can be defined as $\varphi \rightarrow_t {\sim_t}(p_0 \rightarrow_t p_0)$ or as $\varphi \rightarrow_f {\sim_f} (p_0 \rightarrow_f p_0)$, together with the fact that ${\sim_i}\varphi$ is equivalent with ${\sim_t}{\sim_f}\neg\varphi$. □

## 4 Semantic and Syntactic Consequence

We now come to the characterisation of consequence relations. Let us first give semantic definitions.

**Definition 9** Using $\bigsqcap_x$ for greatest lower bound in the $\leq_x$ order ($x \in \{t, f, i\}$) and $\bigsqcup_x$ for least upper bound, let the relations $\models_t$, $\models_f$, $\models_i$, and $\models$ between sets of sentences be defined in the following way.

$$\begin{array}{lll}
\Gamma \models_t \Delta \iff \bigsqcap_{t\,\gamma\in\Gamma} V(\gamma) \leq_t \bigsqcup_{t\,\delta\in\Delta} V(\delta) & \text{for all valuations } V \\
\Gamma \models_f \Delta \iff \bigsqcap_{f\,\gamma\in\Gamma} V(\gamma) \leq_f \bigsqcup_{f\,\delta\in\Delta} V(\delta) & \text{for all valuations } V \\
\Gamma \models_i \Delta \iff \bigsqcap_{i\,\gamma\in\Gamma} V(\gamma) \leq_i \bigsqcup_{i\,\delta\in\Delta} V(\delta) & \text{for all valuations } V \\
\Gamma \models \Delta \iff \Gamma \models_t \Delta \text{ and } \Gamma \models_f \Delta &
\end{array}$$

We have thrown the relation $\models$ into the mix because it models a constraint on reasoning that not only rejects loss of truth but also rejects increase in falsity when going from premises to conclusions. Requiring a computer network to preserve both truth and nonfalsity while drawing conclusions seems a reasonable demand.

The definition of $\models_i$ is just a fancy way of expressing that $\Gamma \models_i \Delta$ if and only if $\bigcap_{\gamma\in\Gamma} V(\gamma) \subseteq \bigcup_{\delta\in\Delta} V(\delta)$, for all $V$, but we have put it in the form above in order to stress the uniformity of the three definitions. It is worthwile, however, to decompose the $\models_t$ and $\models_f$ entailment relations a bit further.

**Proposition 6** $\Gamma \models_t \Delta$ *holds iff the conjunction of the following four statements holds for all valuations* $V$.

$$\begin{array}{rcl}
\mathbf{T} \in V(\gamma) \text{ for all } \gamma \in \Gamma & \Longrightarrow & \mathbf{T} \in V(\delta) \text{ for some } \delta \in \Delta \\
\mathbf{B} \in V(\gamma) \text{ for all } \gamma \in \Gamma & \Longrightarrow & \mathbf{B} \in V(\delta) \text{ for some } \delta \in \Delta \\
\mathbf{F} \in V(\delta) \text{ for all } \delta \in \Delta & \Longrightarrow & \mathbf{F} \in V(\gamma) \text{ for some } \gamma \in \Gamma \\
\mathbf{N} \in V(\delta) \text{ for all } \delta \in \Delta & \Longrightarrow & \mathbf{N} \in V(\gamma) \text{ for some } \gamma \in \Gamma
\end{array}$$

$\Gamma \models_f \Delta$*, on the other hand, is true iff the conjunction of the following four statements holds for all* $V$.

$$\begin{array}{rcl}
\mathbf{T} \in V(\gamma) \text{ for all } \gamma \in \Gamma & \Longrightarrow & \mathbf{T} \in V(\delta) \text{ for some } \delta \in \Delta \\
\mathbf{B} \in V(\delta) \text{ for all } \delta \in \Delta & \Longrightarrow & \mathbf{B} \in V(\gamma) \text{ for some } \gamma \in \Gamma \\
\mathbf{F} \in V(\delta) \text{ for all } \delta \in \Delta & \Longrightarrow & \mathbf{F} \in V(\gamma) \text{ for some } \gamma \in \Gamma \\
\mathbf{N} \in V(\gamma) \text{ for all } \gamma \in \Gamma & \Longrightarrow & \mathbf{N} \in V(\delta) \text{ for some } \delta \in \Delta
\end{array}$$

*Proof* By inspection, using Propositions 1 and 2. □

These decompositions inspire us to define four auxiliary entailment relations.

**Definition 10** For each $x \in \{\mathbf{T}, \mathbf{B}, \mathbf{F}, \mathbf{N}\}$, define the auxiliary entailment relation $\models_x$ by

$$\Gamma \models_x \Delta \Longleftrightarrow x \in \bigcap_{\gamma \in \Gamma} V(\gamma) \Rightarrow x \in \bigcup_{\delta \in \Delta} V(\delta), \quad \text{for all valuations } V$$

Note that no pair of these relations or their inverses are equivalent on sets of premises and conclusions from $\mathscr{L}_{tfi}$.

**Proposition 7** *The relations* $\models_\mathbf{T}$, $\models_\mathbf{F}$, $\models_\mathbf{N}$, $\models_\mathbf{B}$, *and their inverses are pairwise distinct on* $\mathscr{L}_{tfi}$.

*Proof* None of these relations is equivalent with any of their inverses, for $\varnothing \models_x \mathsf{nftb}$ for all $x \in \{\mathbf{T}, \mathbf{B}, \mathbf{F}, \mathbf{N}\}$ while $\mathsf{nftb} \models_x \varnothing$ holds for none. Note further that $\mathsf{nftb} \models_x \mathsf{n}$ holds for $x = \mathbf{N}$, but not for any other of the three values for $x$, so that $\models_\mathbf{N}$ is different from $\models_\mathbf{T}$, $\models_\mathbf{F}$, and $\models_\mathbf{B}$. Showing that other relations $\models_x$ and $\models_y$ are distinct goes in an entirely similar way. □

Proposition 7 is in stark contrast with what is the case in less expressive logics, in which not all elements of **16** can be named. See Shramko & Wansing [14, Lemma 4.2]. A similar situation obtains with respect to Belnap's original four-valued logic versus a functionally complete extension (see Muskens [8]).

Given our definitions thus far the following equivalences between our main consequence relations and certain combinations of the auxiliary ones obviously hold.

$$\begin{aligned}
\Gamma \models_t \Delta &\iff \Gamma \models_{\mathbf{T}} \Delta \text{ and } \Gamma \models_{\mathbf{B}} \Delta \text{ and } \Delta \models_{\mathbf{F}} \Gamma \text{ and } \Delta \models_{\mathbf{N}} \Gamma\\
\Gamma \models_f \Delta &\iff \Gamma \models_{\mathbf{T}} \Delta \text{ and } \Delta \models_{\mathbf{B}} \Gamma \text{ and } \Delta \models_{\mathbf{F}} \Gamma \text{ and } \Gamma \models_{\mathbf{N}} \Delta\\
\Gamma \models_i \Delta &\iff \Gamma \models_{\mathbf{T}} \Delta \text{ and } \Gamma \models_{\mathbf{B}} \Delta \text{ and } \Gamma \models_{\mathbf{F}} \Delta \text{ and } \Gamma \models_{\mathbf{N}} \Delta
\end{aligned}$$

The idea now is to define four syntactic entailment relations $\vdash_x$ $(x \in \{\mathbf{T}, \mathbf{B}, \mathbf{F}, \mathbf{N}\})$, each one characterising its semantic counterpart $\models_x$. In this way the relations $\models_t$, $\models_f$, $\models_i$, and $\models$ will also be provided with syntactic characterisations. We will do this with the help of the signed tableau calculus $\mathbf{PL_{16}}$ (the name stands for '**16**-valued propositional logic'). This calculus will make use of eight signs t, f, n, b, $\not{t}$, $\not{f}$, $\not{n}$, and $\not{b}$, whose informal interpretations will become clear shortly. A *signed* $\mathscr{L}_{tfi}$ *sentence* will be a pair $\mathsf{x} : \varphi$, where $\mathsf{x}$ is one of these signs and $\varphi$ is an $\mathscr{L}_{tfi}$ sentence. In Table 1 nine rule schemes are given which form the heart of our calculus. Note that each of the three conjunctions comes with two rule schemes, each allowing four instantiations of a variable $\mathsf{x}$ ranging over signs, while negations come with one rule scheme each and can each be instantiated in eight different ways. The rule schemes thus sum up eight rules for each connective. Conjunction rules do not change the signs of formulas, but negation rules always do.

For formal considerations it will be useful to have a general form for rules, for which we choose $\vartheta / B_1, \ldots, B_n$, where $\vartheta$ is a signed sentence called the *top formula* of the rule and each $B_i$ is a set of signed sentences called a *set of bottom formulas* of the rule. For example, one instantiation of $(\wedge_f^2)$ could formally be written as $\not{n} : \varphi \wedge_f \psi \;/\; \{\not{n} : \varphi\}, \{\not{n} : \psi\}$ and one instantiation of the $(\wedge_i^1)$ rule could be expressed as $\mathsf{f} : \varphi \wedge_i \psi \;/\; \{\mathsf{f} : \varphi,\ \mathsf{f} : \psi\}$. This general form is useful, even though neither the number of sets of bottom formulas nor their cardinality ever exceeds 2.

Tableaux will be certain sets of *branches*. Let us study the latter and some of the syntactic and semantic properties of notions connected with them before we define the tableaux themselves.

**Table 1** Tableau expansion rules for $\mathbf{PL_{16}}$

| | | |
|---|---|---|
| $\dfrac{\mathsf{x} : \varphi \wedge_t \psi}{\mathsf{x} : \varphi,\ \mathsf{x} : \psi}\ (\wedge_t^1)$ $\mathsf{x} \in \{\not{n}, \not{f}, t, b\}$ | $\dfrac{\mathsf{x} : \varphi \wedge_t \psi}{\mathsf{x} : \varphi \mid \mathsf{x} : \psi}\ (\wedge_t^2)$ $\mathsf{x} \in \{n, f, \not{t}, \not{b}\}$ | $\dfrac{\mathsf{x} : {\sim_t}\, \varphi}{\mathsf{y} : \varphi}\ ({\sim_t})$ $\langle \mathsf{x}, \mathsf{y}\rangle$ or $\langle \mathsf{y}, \mathsf{x}\rangle \in \{\langle n, t\rangle, \langle f, b\rangle, \langle \not{n}, \not{t}\rangle, \langle \not{f}, \not{b}\rangle\}$ |
| $\dfrac{\mathsf{x} : \varphi \wedge_f \psi}{\mathsf{x} : \varphi,\ \mathsf{x} : \psi}\ (\wedge_f^1)$ $\mathsf{x} \in \{n, \not{f}, t, \not{b}\}$ | $\dfrac{\mathsf{x} : \varphi \wedge_f \psi}{\mathsf{x} : \varphi \mid \mathsf{x} : \psi}\ (\wedge_f^2)$ $\mathsf{x} \in \{\not{n}, f, \not{t}, b\}$ | $\dfrac{\mathsf{x} : {\sim_f}\, \varphi}{\mathsf{y} : \varphi}\ ({\sim_f})$ $\langle \mathsf{x}, \mathsf{y}\rangle$ or $\langle \mathsf{y}, \mathsf{x}\rangle \in \{\langle n, f\rangle, \langle t, b\rangle, \langle \not{n}, \not{f}\rangle, \langle \not{t}, \not{b}\rangle\}$ |
| $\dfrac{\mathsf{x} : \varphi \wedge_i \psi}{\mathsf{x} : \varphi,\ \mathsf{x} : \psi}\ (\wedge_i^1)$ $\mathsf{x} \in \{n, f, t, b\}$ | $\dfrac{\mathsf{x} : \varphi \wedge_i \psi}{\mathsf{x} : \varphi \mid \mathsf{x} : \psi}\ (\wedge_i^2)$ $\mathsf{x} \in \{\not{n}, \not{f}, \not{t}, \not{b}\}$ | $\dfrac{\mathsf{x} : {\sim_i}\, \varphi}{\mathsf{y} : \varphi}\ ({\sim_i})$ $\langle \mathsf{x}, \mathsf{y}\rangle$ or $\langle \mathsf{y}, \mathsf{x}\rangle \in \{\langle n, \not{b}\rangle, \langle f, \not{t}\rangle, \langle \not{n}, b\rangle, \langle \not{f}, t\rangle\}$ |

**Definition 11** A (tableau) *branch* is a set of signed sentences. A branch is *closed* if it contains signed sentences $\mathsf{x} : \varphi$ and $\not\mathsf{x} : \varphi$ for $\mathsf{x} \in \{\mathsf{n}, \mathsf{f}, \mathsf{t}, \mathsf{b}\}$. A branch that is not closed is called *open*.

If $\mathscr{B}$ is a branch and $\mathsf{x} : \varphi \in \mathscr{B}$ is a signed sentence, then $\mathsf{x} : \varphi$ is *fulfilled* in $\mathscr{B}$ if (a) $\varphi$ is atomic, or (b) $\varphi$ is complex and $B_i \subseteq \mathscr{B}$ for one set of bottom formulas of the unique rule $\mathsf{x} : \varphi / B_1, \ldots, B_n$. A branch $\mathscr{B}$ is *completed* if $\mathscr{B}$ is closed or $\vartheta$ is fulfilled in $\mathscr{B}$ for every $\vartheta \in \mathscr{B}$.

Satisfiability of branches will be a central notion. We define it as follows.

**Definition 12** Let $\Theta$ be a set of signed $\mathscr{L}_{tfi}$ sentences and let $V$ be an $\mathscr{L}_{tfi}$ valuation. We say that $V$ *satisfies* $\Theta$ iff the following statements hold.

$$\begin{array}{llll}
\mathsf{t} : \varphi \in \Theta & \Rightarrow \mathbf{T} \in V(\varphi) & \not\mathsf{t} : \varphi \in \Theta & \Rightarrow \mathbf{T} \notin V(\varphi) \\
\mathsf{f} : \varphi \in \Theta & \Rightarrow \mathbf{F} \in V(\varphi) & \not\mathsf{f} : \varphi \in \Theta & \Rightarrow \mathbf{F} \notin V(\varphi) \\
\mathsf{n} : \varphi \in \Theta & \Rightarrow \mathbf{N} \in V(\varphi) & \not\mathsf{n} : \varphi \in \Theta & \Rightarrow \mathbf{N} \notin V(\varphi) \\
\mathsf{b} : \varphi \in \Theta & \Rightarrow \mathbf{B} \in V(\varphi) & \not\mathsf{b} : \varphi \in \Theta & \Rightarrow \mathbf{B} \notin V(\varphi)
\end{array}$$

So the signs wear their interpretations on their sleeves. We also say that $V$ *satisfies* $\vartheta$ if $V$ satisfies $\{\vartheta\}$ and that a signed sentence or a set of signed sentences is *satisfiable* if some $V$ satisfies it.

Satisfaction is preserved from the top formula in a rule to one of its sets of bottom formulas and vice versa, as the following proposition attests.

**Proposition 8** *For any instantiation of a rule scheme in Table 1, a valuation satisfies the top formula of the rule if and only if it satisfies a set of bottom formulas of the rule.*

*Proof* By inspection, using Proposition 2 and Definition 3. □

It is immediate from the definitions that a branch that is satisfiable cannot be closed. The following also holds.

**Lemma 1** *A branch that is completed and open is satisfiable.*

*Proof* Let $\mathscr{B}$ be completed and open and, for all propositional constants $p$, let

$$\begin{aligned} V(p) = \{x \mid (x = \mathbf{T} \text{ and } \mathsf{t} : p \in \mathscr{B}) \text{ or } (x = \mathbf{F} \text{ and } \mathsf{f} : p \in \mathscr{B}) \text{ or} \\ (x = \mathbf{N} \text{ and } \mathsf{n} : p \in \mathscr{B}) \text{ or } (x = \mathbf{B} \text{ and } \mathsf{b} : p \in \mathscr{B})\} \end{aligned}$$

We prove by induction on formula complexity that $V$ satisfies all signed sentences $\mathsf{x} : \varphi \in \mathscr{B}$. If $\varphi$ is a propositional constant, then $V$ satisfies $\mathsf{x} : \varphi$ by the definition just given and the fact that $\mathscr{B}$ is open. So, let $\mathsf{x} : \varphi \in \mathscr{B}$ for some sign $\mathsf{x}$ and complex formula $\varphi$. Since $\mathscr{B}$ is completed, a set of bottom formulas $B$ of the unique rule of which $\mathsf{x} : \varphi$ is the top formula is a subset of $\mathscr{B}$. But, for all signed sentences $\mathsf{y} : \psi \in B$, $\psi$ is a subformula of $\varphi$, so that, by induction, $V$ satisfies $\mathsf{y} : \psi$. So $V$ satisfies $B$, and by Proposition 8, $V$ satisfies $\mathsf{x} : \varphi$. □

Let us now define tableaux.

**Definition 13** Let $\mathscr{T}$ and $\mathscr{T}'$ be sets of branches. We say that $\mathscr{T}'$ is a *one-step expansion* of $\mathscr{T}$ if, for some $\mathscr{B} \in \mathscr{T}$, $\vartheta \in \mathscr{B}$, and rule $\vartheta / B_1, \ldots, B_n$, $\mathscr{T}' = (\mathscr{T} \backslash \{\mathscr{B}\}) \cup \{\mathscr{B} \cup B_1, \ldots, \mathscr{B} \cup B_n\}$.

Let $\mathscr{B}$ be a finite branch. A set of branches $\mathscr{T}$ is a *tableau with initial branch* $\mathscr{B}$ if there is a sequence $\mathscr{T}_0, \mathscr{T}_1, \ldots, \mathscr{T}_n$ such that $\mathscr{T}_0 = \{\mathscr{B}\}$, $\mathscr{T}_n = \mathscr{T}$, and each $\mathscr{T}_{i+1}$ is a one-step expansion of $\mathscr{T}_i$ $(0 \leq i < n)$. We also say that a finite $\mathscr{B}$ *has* tableau $\mathscr{T}$ if $\mathscr{T}$ is a tableau with initial branch $\mathscr{B}$. A tableau $\mathscr{T}$ is *open* if some $\mathscr{B} \in \mathscr{T}$ is open, otherwise $\mathscr{T}$ is *closed*. A tableau is *completed* if each of its branches is completed.

**Proposition 9** *(a) Every finite branch has a tableau that is completed. (b) Let $\mathscr{T}$ be a tableau with initial branch $\mathscr{B}$. If $\mathscr{B}$ is satisfiable then there is a satisfiable branch $\mathscr{B}'$ such that $\mathscr{B} \subseteq \mathscr{B}' \in \mathscr{T}$.*

*Proof* The (a) part is proved just as in classical propositional logic. For the (b) part, consider the sequence $\{\mathscr{B}\} = \mathscr{T}_0, \mathscr{T}_1, \ldots, \mathscr{T}_n = \mathscr{T}$ that must exist according to the previous definition and do an induction on $n$, using Proposition 8. □

With the help of this Proposition and Lemma 1 the following lemma now follows immediately.

**Lemma 2** *A finite set of signed sentences is unsatisfiable iff it has a closed tableau.*

*Proof* Suppose the finite set of signed sentences $\mathscr{B}$ is unsatisfiable. By the previous proposition $\mathscr{B}$ has a completed tableau $\mathscr{T}$. Since $\mathscr{B}$ is included in every branch of $\mathscr{T}$, no such branch can be satisfiable. By Lemma 1 it follows that $\mathscr{T}$ must be closed. Conversely, assume that $\mathscr{B}$ is satisfiable. Then $\mathscr{B}$ has a completed tableau $\mathscr{T}$ with at least one satisfiable branch, so that $\mathscr{T}$ cannot be closed. □

With the help of our tableaux we define four auxiliary syntactic consequence relations.

**Definition 14** Let $\Gamma$ and $\Delta$ be finite sets of $\mathscr{L}_{tfi}$ sentences and define

$$\begin{aligned}
\Gamma \vdash_{\mathbf{T}} \Delta &:= \{\mathsf{t} : \varphi \mid \varphi \in \Gamma\} \cup \{\not{\mathsf{t}} : \varphi \mid \varphi \in \Delta\} \text{ has a closed tableau;}\\
\Gamma \vdash_{\mathbf{F}} \Delta &:= \{\mathsf{f} : \varphi \mid \varphi \in \Gamma\} \cup \{\not{\mathsf{f}} : \varphi \mid \varphi \in \Delta\} \text{ has a closed tableau;}\\
\Gamma \vdash_{\mathbf{N}} \Delta &:= \{\mathsf{n} : \varphi \mid \varphi \in \Gamma\} \cup \{\not{\mathsf{n}} : \varphi \mid \varphi \in \Delta\} \text{ has a closed tableau;}\\
\Gamma \vdash_{\mathbf{B}} \Delta &:= \{\mathsf{b} : \varphi \mid \varphi \in \Gamma\} \cup \{\not{\mathsf{b}} : \varphi \mid \varphi \in \Delta\} \text{ has a closed tableau.}
\end{aligned}$$

Using Lemma 2 it immediately follows that these provide sound and complete characterisations of the auxiliary semantic entailment relations defined earlier.

**Lemma 3** $\Gamma \models_x \Delta \Longleftrightarrow \Gamma \vdash_x \Delta$, *for each* $x \in \{\mathbf{T}, \mathbf{B}, \mathbf{F}, \mathbf{N}\}$.

*Proof* Let $\mathscr{B}_{\mathbf{T}} = \{\mathrm{t} : \varphi \mid \varphi \in \Gamma\} \cup \{\not{\mathrm{t}} : \varphi \mid \varphi \in \Delta\}$. Then $\Gamma \models_{\mathbf{T}} \Delta \iff \mathscr{B}_{\mathbf{T}}$ is unsatisfiable $\iff \mathscr{B}_{\mathbf{T}}$ has a closed tableau $\iff \Gamma \vdash_{\mathbf{T}} \Delta$. The other cases are quite similar. □

**Definition 15** Let $\Gamma$ and $\Delta$ be finite sets of $\mathscr{L}_{\mathit{tfi}}$ sentences. Define

$$\begin{array}{rcl}
\Gamma \vdash_t \Delta & \iff & \Gamma \vdash_{\mathbf{T}} \Delta \text{ and } \Gamma \vdash_{\mathbf{B}} \Delta \text{ and } \Delta \vdash_{\mathbf{F}} \Gamma \text{ and } \Delta \vdash_{\mathbf{N}} \Gamma \\
\Gamma \vdash_f \Delta & \iff & \Gamma \vdash_{\mathbf{T}} \Delta \text{ and } \Delta \vdash_{\mathbf{B}} \Gamma \text{ and } \Delta \vdash_{\mathbf{F}} \Gamma \text{ and } \Gamma \vdash_{\mathbf{N}} \Delta \\
\Gamma \vdash_i \Delta & \iff & \Gamma \vdash_{\mathbf{T}} \Delta \text{ and } \Gamma \vdash_{\mathbf{B}} \Delta \text{ and } \Gamma \vdash_{\mathbf{F}} \Delta \text{ and } \Gamma \vdash_{\mathbf{N}} \Delta \\
\Gamma \vdash \Delta & \iff & \Gamma \vdash_t \Delta \text{ and } \Gamma \vdash_f \Delta
\end{array}$$

From Lemma 3 and our definitions the next theorem now immediately follows.

**Theorem 1** *If $\Gamma$ and $\Delta$ are finite sets of $\mathscr{L}_{\mathit{tfi}}$ sentences the following hold.*

$$\begin{array}{ll}
\Gamma \models_t \Delta \iff \Gamma \vdash_t \Delta & \Gamma \models_f \Delta \iff \Gamma \vdash_f \Delta \\
\Gamma \models_i \Delta \iff \Gamma \vdash_i \Delta & \Gamma \models \Delta \iff \Gamma \vdash_t \Delta \textit{ and } \Gamma \vdash_f \Delta
\end{array}$$

This solves the characterisation problem of the logics connected with $\mathscr{L}_{\mathit{tfi}}$. Finite sequents can now be decided with the help of developing several tableaus. In general four tableaus are needed to decide $\models_t$, $\models_f$, or $\models_i$ sequents and six are needed to decide sequents on the basis of $\models$. It may of course happen that fewer tableaux are needed if sequents have a special form. For $\models_t$ sequents from $\mathscr{L}_t$, the language based on $\{\wedge_t, \sim_t\}$, only one tableau needs to be developed, for example, as the others will be isomorphic. This is already clear from Shramko & Wansing's [14, Lemma 4.2], but can also be glossed from our tableau rules.

## 5 Conclusion

We have given an analytic tableau calculus $\mathbf{PL_{16}}$ for the whole of $SIXTEEN_3$, with connectives for all its truth functions. While each connective comes with eight rules and in general several tableaux must be developed in order to check entailments (six tableaux, for instance, in the case of $\models$ entailments), the calculus is still essentially simple, as the many rules can be brought under much fewer rule schemes and the system that results has a remarkable family resemblance to the usual signed tableau calculus for classical propositional logic.[3]

**Acknowledgments** We would like to thank the referee for encouraging words and helpful comments. Stefan Wintein wants to thank the Netherlands Organisation for Scientific Research (NWO) for funding the project *The Structure of Reality and the Reality of Structure* (project leader: F. A. Muller), in which he is employed. Reinhard Muskens gratefully acknowledges NWO's funding of his project 360-80-050, *Towards Logics that Model Natural Reasoning.*

[3]This simplicity has a pay off because it is now very easy to implement the logic. Mainly for their own amusement and instruction the authors have written a simple theorem prover 16$T^AP$, loosely based on (the propositional part of) Beckert & Posegga's [2] lean$T^AP$ marvel. It similarly exploits Prolog's left-to-right depth-first evaluation mechanism, but the code unfortunately is far less concise than Beckert & Posegga's.

## References

1. Baaz, M., Fermüller, C., Zach, R. (1994). Elimination of cuts in first-order finite-valued logics. *Journal of Information Processing and Cybernetics*, *29*, 333–355.
2. Beckert, B., & Posegga, J. (1995). lean$T^AP$: Lean Tableau-based deduction. *Journal of Automated Reasoning*, *15*(3), 339–358. http://web.sec.uni-passau.de/papers/LeanTaP.pdf.
3. Belnap, N.D. (1976). How a computer should think. In G. Ryle (Ed.), *Contemporary aspects of philosophy* (pp. 30–56). Stocksfield: Oriel Press.
4. Belnap, N.D. (1977). A useful four-valued logic. In J. Dunn, & G. Epstein (Eds.), *Modern uses of multiple-valued logic* (pp. 8–37). Dordrecht: Reidel.
5. Kamide, N., & Wansing, H. (2009). Sequent calculi for some trilattice logics. *The Review of Symbolic Logic*, *2*(2), 374–395.
6. Langholm, T. (1996). How different is partial logic? In P. Doherty (Ed.), *Partiality, modality, and nonmonotonicity* (pp. 3–43). Stanford: CSLI.
7. Muskens, R.A. (1995). *Meaning and partiality*. Stanford: CSLI.
8. Muskens, R.A. (1999). On partial and paraconsistent logics. *Notre Dame Journal of Formal Logic*, *40*(3), 352–374.
9. Odintsov, S. (2009). On Axiomatizing Shramko-Wansing's logic. *Studia Logica*, *91*, 407–428.
10. Odintsov, S., & Wansing, H. (2014). The logic of generalized truth values and the logic of bilattices. In *Studia Logica*.
11. Rivieccio, U. (2013). Representation of interlaced trilattices. *Journal of Applied Logic*, *11*, 174–189.
12. Rousseau, G. (1967). Sequents in many-valued logic I. *Fundamenta Mathematicae*, *60*, 23–33.
13. Schröter, K. (1955). Methoden zur Axiomatisierung beliebiger Aussagen- und Prädikatenkalküle. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, *1*, 241–251.
14. Shramko, Y., & Wansing, H. (2005). Some Useful 16-Valued Logics: How a Computer Network Should Think. *Journal of Philosophical Logic*, *34*, 121–153.
15. Shramko, Y., & Wansing, H. (2011). Truth and falsehood: An inquiry into generalized logical values. In *Trends in logic* Vol. 36: Springer.
16. Wansing, H. (2009). The power of Belnap: Sequent systems for $SIXTEEN_3$. *Journal of Philosophical Logic*, *39*, 369–393.
17. Wansing, H. (2012). A non-inferentialist, anti-realistic conception of logical truth and falsity. *Topoi*, *31*, 93–100.
18. Wansing, H., & Kamide, N. (2010). Intuitionistic Trilattice Logics. *Journal of Logic and Computation*, *20*(6), 1201–1229.
19. Wintein, S., & Muskens, R.A. (2012). A calculus for Belnap's logic in which each proof consists of two trees. *Logique & Analyse*, *220*, 643–656.
20. Wintein, S., & Muskens, R.A. (2014). From bi-facial truth to bi-facial proofs. Studia Logica. Online First.
21. Zach, R. (1993). Proof theory of finite-valued logics. Technical Report TUW-E185.2-Z.1-93. Institut für Computersprachen, Technische Universität Wien.