South American Journal of Logic
Vol. 6, n. 2, pp. 325–345,
ISSN: 2446-6719

Analytic Proofs for Logics of Evidence and Truth

Walter Carnielli, Lorenzzo Frade, and Abilio Rodrigues

Dedicated to Francisco Miró Quesada Cantuarias,
the godfather of paraconsistency.

Abstract

This paper presents a sound, complete, and decidable analytic tableau system for the logic of evidence and truth LET_F, introduced in Rodrigues, Bueno-Soler, and Carnielli [19]. LET_F is an extension of the logic of first-degree entailment (FDE), also known as Belnap-Dunn logic. FDE is a widely studied four-valued paraconsistent logic, with applications in computer science and in the algebra of processes. LET_F extends FDE in a very natural way, by adding a classicality operator $\circ$, which recovers classical logic for propositions in its scope, and a non-classicality operator $\bullet$, dual of $\circ$.

Keywords: logics of evidence and truth; analytic tableau proofs; logic of first-degree entailment; paraconsistency; paracompleteness.

1 Introduction

The aim of this paper is to present an analytic tableau system for the logic LET_F, which is a member of a family of logics called *logics of evidence and truth* (LETs) [6, 19]. The motivation for LETs is the idea that, in real-life reasoning, we deal with positive and negative evidence, and such evidence can be conclusive or non-conclusive. LETs thus combine two different notions of logical consequence in the same formal system: one preserves truth (classical consequence), the other preserves evidence—hence, the name 'logics of evidence and truth'. Evidence is thought of as a notion weaker than truth, in the sense that there may be evidence for a proposition A even in the case A is not true. Positive and negative evidence, respectively evidence for truth and for falsity, are independent and non-complementary, and negative evidence for A is identified with positive evidence for $\neg A$. LETs are paraconsistent, since it may be that there is conflicting non-conclusive evidence for a proposition A, and paracomplete, since it may happen that there is no evidence at all for A.[1]

The logic LET_F, introduced in [19], is an extension of the logic of first-degree entailment (FDE), also called Belnap-Dunn logic [2, 10, 16]. LET_F is equipped with a classicality operator $\circ$ and a non-classicality operator $\bullet$ which is the dual of $\circ$. The deductive behavior of $\circ$ and $\bullet$ is given by the following inferences:

(1) $\circ A, \bullet A \vdash B$,
(2) $\vdash \circ A \vee \bullet A$,
(3) $\circ A, A, \neg A \vdash B$ (although $A, \neg A \nvdash B$),
(4) $\circ A \vdash A \vee \neg A$ (although $\nvdash A \vee \neg A$),
(5) $\vdash A \vee \neg A \vee \bullet A$,
(6) $A, \neg A \vdash \bullet A$.

According to the intended interpretation in terms of evidence, items (1) and (2) together mean that either there is or there is not conclusive evidence for A. Items (3) and (4), as in any LET, recover classical negation for propositions in the scope of $\circ$, and (5) and (6) are dual, respectively, of (3) and (4). Due to items (3) and (4), LET_F is a logic of formal inconsistency and undeterminedness [cf. 8, 14].

LET_F, as LETs in general, can also be interpreted in terms of information, which may be unreliable or reliable, the latter being subjected to classical logic [see 19, sec. 2.2.1]. In this case, $\circ A$ means that the information about A, positive or negative, is reliable. In [1], Kripke models for LET_F have been proposed. These models intend to represent a database that receives information as time passes, and such information can be positive, negative, unreliable, or reliable. This idea fits the interpretation of Belnap-Dunn logic as an information-based logic, but adds to the four scenarios expressed by it two new scenarios: reliable information (i) for the truth and (ii) for the falsity of a given proposition. FDE was probably the first logic to be applied in computer science and, more recently, in description logics and in the algebra of processes [see e.g. 3, 11, 17, 20]. An extension of FDE such as LET_F has an ample range of possible applications, especially in Bayesian decision procedures under uncertainty [cf. 4].

This paper extends the investigation on LET_F carried out in [1, 19]. It has been remarked in [19, sect. 3.2] that LET_F is decidable, but a detailed algorithm has not been presented. Our main aim here is to present a sound,

[1] A more detailed account of the notion of evidence that underlies the intended interpretation of LETs can be found in [18].

complete, and decidable tableaux system for LET_F.[2] The tableaux to be proposed are analytic because the formulas yielded by the application of a rule to a formula F are always less complex than F, and counterexamples can be obtained from open branches of terminated tableaux.

The remainder of this paper is structured as follows. Section 2 presents the valuation semantics of LET_F, and section 3, the corresponding tableau system. In section 4 we prove the soundness and completeness of the LET_F-tableau system with respect to the semantics of section 2, and also that the analytic tableau system provides a decision procedure for LET_F. Finally, in section 5, some examples of LET_F-tableaux are given and commented.

2 Valuation Semantics for LET_F

The language $\mathcal{L}$ of LET_F is composed of denumerably many sentential letters $p_1, p_2, \ldots$, the unary connectives $\circ$, $\bullet$, and $\neg$, the binary connectives $\wedge$ and $\vee$, and parentheses. The set of formulas of $\mathcal{L}$, which is also denoted by $\mathcal{L}$, is inductively defined in the usual way. Roman capitals $A, B, C, \ldots$ will be used as metavariables for the formulas of $\mathcal{L}$, and Greek capitals $\Gamma, \Delta, \Sigma, \ldots$ as metavariables for sets of formulas $\mathcal{L}$.

In [19], a natural deduction system was presented for LET_F together with the following sound and complete semantics.

Definition 1. [Valuation semantics for LET_F]

A valuation semantics for LET_F is a collection of LET_F-valuations defined as follows. A function $v : \mathcal{L} \to \{0, 1\}$ is a LET_F-valuation if it satisfies the following clauses:

(v1) $v(A \wedge B) = 1$ iff $v(A) = 1$ and $v(B) = 1$,
(v2) $v(A \vee B) = 1$ iff $v(A) = 1$ or $v(B) = 1$,
(v3) $v(\neg(A \wedge B)) = 1$ iff $v(\neg A) = 1$ or $v(\neg B) = 1$,
(v4) $v(\neg(A \vee B)) = 1$ iff $v(\neg A) = 1$ and $v(\neg B) = 1$,
(v5) $v(A) = 1$ iff $v(\neg\neg A) = 1$,
(v6) If $v(\circ A) = 1$, then $v(A) = 1$ if and only if $v(\neg A) = 0$,
(v7) $v(\bullet A) = 1$ iff $v(\circ A) = 0$.

Definition 2. We say that a formula A is a semantical consequence of Γ, $\Gamma \vDash A$, if and only if, for every valuation v, if $v(B) = 1$ for all $B \in \Gamma$, then $v(A) = 1$.

[2] Analytic tableaux for some logics of formal inconsistency have already been proposed in [5] and [7, sec. 3.5].

The semantics above is non-deterministic in the sense that the semantic value of complex formulas is not always functionally determined by its parts, as the following non-deterministic matrix (also called quasi-matrix) shows.[3]

Example 3. (i) $\circ p \vDash p \vee \neg p$,

(ii) $p \vee \neg p \nvDash \circ p$,

(iii) $\nvDash \circ p \vee \neg \circ p$.

1	p	0						1					
2	$\neg p$	0		1				0				1	
3	$p \vee \neg p$	0		1				1				1	
4	$\circ p$	0		0		1		0		1		0	
5	$\neg \circ p$	0	1	0	1	0	1	0	1	0	1	0	1
		v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}

There is no valuation v such that $v(\circ p) = 1$ and $v(p \vee \neg p) = 0$, so (i) holds. Valuation v_3 provides a counterexample to both (ii) and (iii). Note that lines 2, 4, and 5 bifurcate: line 2 because p and $\neg p$ are completely independent of each order, line 4 because there are no sufficient conditions for $v(\circ A) = 1$, and line 5 for the same reason as line 2. Indeed, a feature of LET_F (as well as of some *LFI*s) is that when $\circ A$ ($\bullet A$) occurs in the scope of $\neg$, unless $\circ\circ A$ ($\circ\bullet A$) holds, the value of $\neg\circ A$ ($\neg\bullet A$) is not functionally determined by the value of $\circ A$ ($\bullet A$). The negation in these formulas is still a weak negation.

3 An Analytic Tableau System for LET_F

Given the semantics above, we shall prove in section 4 that the following tableau rules constitute a sound, complete, and decidable proof system for LET_F. We will consider informally here the usual notions related to tableaux: trees, branches, nodes, etc. The labels 0 and 1 refer to metamathematical markers, intuitively related to the semantic values *0* and *1*.

Definition 4. [Tableau rules for LET_F]

Rule 1

$1(A \wedge B)$
$1(A)$
$1(B)$

Rule 2

$0(A \wedge B)$

$0(A)$ | $0(B)$

[3] On non-deterministic valuation semantics, see Loparic [12, 13].

Rule 3

$$1(\neg(A \wedge B))$$
$$1(\neg A) \qquad | \qquad 1(\neg B)$$

Rule 4

$$0(\neg(A \wedge B))$$
$$0(\neg A)$$
$$0(\neg B)$$

Rule 5

$$1(A \vee B)$$
$$1(A) \qquad | \qquad 1(B)$$

Rule 6

$$0(A \vee B)$$
$$0(A)$$
$$0(B)$$

Rule 7

$$1(\neg(A \vee B))$$
$$1(\neg A)$$
$$1(\neg B)$$

Rule 8

$$0(\neg(A \vee B))$$
$$0(\neg A) \qquad | \qquad 0(\neg B)$$

Rule 9

$$1(\neg\neg A)$$
$$1(A)$$

Rule 10

$$0(\neg\neg A)$$
$$0(A)$$

Rule 11

$$1(\circ A)$$
$$1(A) \qquad | \qquad 0(A)$$
$$0(\neg A) \qquad | \qquad 1(\neg A)$$

Rule 12

$$1(\bullet A)$$
$$0(\circ A)$$

Rule 13

$$0(\bullet A)$$
$$1(\circ A)$$

There is no need for a rule for $0(\circ A)$. Such a rule, call it **R**, would conclude $1(\bullet A)$ from $0(\circ A)$. Besides yielding a loop with **Rule 12**, it can be shown that **R** is not necessary at all. Suppose the application of **R** to $0(\circ A)$ yielded a closed branch b such that both $1(\bullet A)$ and $0(\bullet A)$ occur in b. But in this case, it would be enough to apply **Rule 13** to $0(\bullet A)$, obtaining a branch b' containing $1(\circ A)$ and $0(\circ A)$, and b' would be a closed branch.

Concerning **Rule 11**, recall that the symbol $\circ$ in LET_F expresses *classicality*, i.e., a formula $\circ A$ implies that A behave classically. The classical behavior of A is recovered by recovering classical negation for A: either A, or $\neg A$ holds, and not both. This is precisely what **Rule 11** does. The semantic clause for $\circ A$ has only a necessary condition for $v(\circ A) = 1$, and the absence of a rule for $0(\circ A)$ mimics this fact.

Moreover, note that there are no tableau rules for $\neg\circ A$ and $\neg\bullet A$. As remarked in Example 3 above, the value of $\neg\circ A$ ($\neg\bullet A$) is not functionally determined by the value of $\circ A$ ($\bullet A$), except when $\circ\circ A$ ($\circ\bullet A$) holds. This is illustrated by the fact that in LET_F neither $\vdash \circ A \vee \neg\circ A$, nor $\circ A, \neg\circ A \vdash B$ hold (we will see the respective tableaux in section 5 below).

Definition 5. [LET_F tableaux]

1. We define a *tableau* for a set Δ of signed formulas as a tree whose first node contains all the signed formulas in Δ, and whose subsequent nodes are obtained by applications of the tableau rules given in Definition 4.
2. A tableau branch is *closed* if it contains a pair of signed formulas $1(F)$ and $0(F)$. If a branch is not closed, we say it is open. A tableau is closed if all its branches are closed.
3. When no rule can be applied to any open branch, the tableau is *terminated.* Every closed tableau is also a terminated tableau.
4. A closed tableau is *pruned* if all formulas in Δ that have not been used in its branches are deleted.
5. A formula A has a *proof* from premisses Γ, denoted $\Gamma \vdash A$, if there is a closed tableau for the set $\{1(B) : B \in \Gamma\} \cup \{0(A)\}$. Γ can be empty, and in this case a proof of $\vdash A$ reduces to a tableau for the singleton $\{0(A)\}$.

Theorem 6. [Compacteness of tableau proofs]

$\Gamma \vdash A$ iff $\Gamma_0 \vdash A$, for $\Gamma_0 \subset \Gamma$, Γ_0 finite.

Proof. First, note that a pruned closed tableau is finite, since any tableau is a finite collection of branches and every closed branch is finite. So, after a pruning procedure, only a finite subset $\Gamma_0 \subset \Gamma$ has been used (that is, the pruning procedure deletes any possibly infinite collection of sentences of Γ not used in the process of tableau closure). □

An analytic tableau is a procedure of reductio ad absurdum whose aim is to obtain a closed tableau, and thus proving $\Gamma \vdash A$. A closed tableau is a halting condition of a decision procedure, and in the case of LET_F this halting condition is always obtained, since every LET_F-tableau terminates (cf. Theorem 8 below). If the tableau is open, an open branch gives a valuation that is a counterexample for $\Gamma \vdash A$. We will see that LET_F-tableaux constitute an elegant decision procedure for LET_F.

4 Soundness and Completeness

In this section we prove soundness, completeness, and decidability of LET_F-tableaux. We start by defining the complexity $\mathcal{C}$ of a formula in the language of LET_F.

Definition 7. [Complexity]

The *complexity* of a formula F of the language of LET_F is given by the map $\mathcal{C} : F \rightarrow \mathbb{N}$, such that:

1. $\mathcal{C}(p) = 0$;
2. $\mathcal{C}(\neg A) = \mathcal{C}(A) + 1$;
3. $\mathcal{C}(A * B) = \mathcal{C}(A) + \mathcal{C}(B) + 1$, for $* \in \{\wedge, \vee\}$;
4. $\mathcal{C}(\circ A) = \mathcal{C}(A) + 2$;
5. $\mathcal{C}(\bullet A) = \mathcal{C}(A) + 3$.

4.1 Soundness

Theorem 8. [Termination]

Every LET_F-tableau terminates: after a finite number of steps no more rules can be applied.

Proof. The result follows from the fact that each rule results in formulas with less complexity or formulas to which no rule can be applied. □

Definition 9. [LET_F-satisfiable branch]

A branch b is *LET_F-satisfiable* if there is a LET_F-valuation v such that for every formula that occurs in b:

1. If $1(F)$ is in b, then $v(F) = 1$,
2. If $0(F)$ is in b, then $v(F) = 0$.

In this case, we say that the valuation v satisfies the branch b.

Clearly, if a branch is closed, it cannot be satisfiable, because there is no valuation such that $v(F) = 1$ and $v(F) = 0$. It is worth noting that a LET_F-satisfiable branch, as expected, differs from a satisfiable branch in classical logic. A LET_F-branch b may be satisfiable even if $1(F)$ and $1(\neg F)$, as well as $0(F)$ and $0(\neg F)$, are both in b. The condition for a branch b of a LET_F-tableau to be closed (cf. Definition 5 item 2) is that for some F, $1(F)$ and $0(F)$ are in b. Moreover, although there is no LET_F-valuation such that $v(F) = 1$, $v(\neg F) = 1$, and $v(\circ F) = 1$, and no open terminated branch b can be such that $1(F)$, $1(\neg F)$, and $1(\circ F)$ are all in b, we do not need to add the latter as a condition for a closed branch. If $1(\circ F)$ is in b, **Rule 11** must be applied, and b will bifurcate into two branches: b', with $1(F)$ and $0(\neg F)$, and b'', with $0(F)$ and $1(\neg F)$. If $1(F)$ and $1(\neg F)$ are both in b, then both branches b' and b'' are closed.

Lemma 10. [Satisfiable branches]

(i) If a non-branching rule is applied to a LET_F-satisfiable branch, the result is another LET_F-satisfiable branch.

(ii) If a branching rule is applied to a LET_F-satisfiable branch, at least one of the resulting branches is also a LET_F-satisfiable branch.

Proof.
To prove (i), we have to show that every non-branching rule applied to a satisfiable branch results in another satisfiable branch.

For **Rule 1**, suppose b is a satisfiable branch containing $1(A \wedge B)$. Therefore, by Definition 9, there is a valuation v such that v satisfies b and $v(A \wedge B) = 1$. Now, applying **Rule 1** to b yields a branch b' such that $1(A)$ and $1(B)$, as well as $1(A \wedge B)$, are in b'. By Definition 1, since $v(A \wedge B) = 1$, we have that $v(A) = 1$ and $v(B) = 1$. Therefore, b' is a satisfiable branch. Analogous reasoning applies to **Rule 4**, **Rule 6**, **Rule 7**, **Rule 9**, and **Rule 10**.

For **Rule 12**, suppose b is a satisfiable branch containing $1(\bullet A)$. By Definition 9, there is a valuation v such that $v(\bullet A) = 1$, and by Definition 1, $v(\circ A) = 0$. An application of **Rule 12** yields a branch b' such that $0(\circ A)$ is in b', and v satisfies b'. Analogous reasoning applies to **Rule 13**.

To prove (ii), we must show that every branching rule applied to a satisfiable branch results in at least one satisfiable branch.

For **Rule 2**. Suppose b is a satisfiable branch containing $0(A \wedge B)$. Thus there is a valuation v such that v satisfies b and either $v(A) = 0$ or $v(B) = 0$. An application of **Rule 2** yields two branches: b' and b'', such that $0(A)$ occurs in b' and $0(B)$ in b''. Therefore, either v satisfies b' or v satisfies b''. Analogous reasoning applies to **Rule 3**, **Rule 5**, and **Rule 8**. For **Rule 11**, suppose b is a satisfiable branch containing $1(\circ A)$. Thus there is a valuation v, such that v satisfies b and $v(\circ A) = 1$. Then by Definition 1 either $v(A) = 1$ and

$v(\neg A) = 0$, or $v(A) = 0$ and $v(\neg A) = 1$. Now, if we apply **Rule 11** to b, we get two branches: b' with $1(A)$ and $0(\neg A)$, and b'' with $0(A)$ and $1(\neg A)$. Therefore, either v satisfies b', or v satisfies b''. □

Theorem 11. [Soundness]

If $\Gamma \vdash A$, then $\Gamma \vDash A$.

Proof. Assume $\Gamma \vdash A$ and suppose for reductio that $\Gamma \nvDash A$. Let b be the first node of the tableau for $\Gamma \vdash A$, with every formula F from Γ labeled with 1 ($1(F_1)$, $1(F_2)$, etc.) and $0(A)$. It follows from $\Gamma \nvDash A$ that there is a LET_F-valuation v such that for every formula F from Γ, $v(F) = 1$, and $v(A) = 0$. Therefore, by Definition 9, b is satisfiable. But from Lemma 10 we have that if any rule is applied to b, at least one of the resulting branches will be satisfiable. Hence, after a finite number of rule applications, the tableau for $\Gamma \vdash A$ terminates with (at least) one satisfiable branch b'. There is thus a valuation v' such that for every formula F that occurs in b', if $1(F)$ is in b', $v(F) = 1$, and if $0(F)$ is in b', $v(F) = 0$, and b' cannot contain any formula F such that $1(F)$ and $0(F)$ are both in b' – otherwise, v' would not be a valuation. But then, b' is an open branch, and $\Gamma \nvdash A$, which contradicts the initial assumption. Therefore, $\Gamma \vDash A$. □

4.2 Completeness

In order to prove completeness of LET_F-tableaux with respect to the valuation semantics of LET_F, we show the contrapositive: if $\Gamma \nvdash A$, then $\Gamma \nvDash A$. $\Gamma \nvdash A$ just in case there is an open branch in the tableau. Let b be this open branch. We have to show that there is a valuation v induced by b such that for every formula F from Γ, $v(F) = 1$, and $v(A) = 0$. We therefore begin by defining a valuation induced by an open branch.

Definition 12. [Semi-valuation induced by an open branch]

Let a *literal* be a propositional letter or the negation of a propositional letter. Let b be an open branch of a terminated tableau. The semi-valuation s induced by b is such that:

1. For every literal l such that $1(l)$ is in b, $s(l) = 1$;
2. For every literal l such that $1(l)$ is not in b, $s(l) = 0$;
3. If $1(\circ A)$ is in b, then $s(\circ A) = 1$ and $s(\bullet A) = 0$;
4. If $1(\circ A)$ is not in b, then $s(\circ A) = 0$ and $s(\bullet A) = 1$;
5. $s(\neg\circ A) = 1$ if, and only if, $1(\neg\circ A)$ is in b;
6. $s(\neg\bullet A) = 1$ if, and only if, $1(\neg\bullet A)$ is in b.

As remarked in Example 3, the semantic values of $\neg\circ A$ and $\neg\bullet A$ are not functionally determined by the values of $\circ A$ and $\bullet A$. It is for this reason that the items 5 and 6 in Definition 13 above have to be explicitly given.

Lemma 13. [Valuation induced by an open branch]

Let b be an open branch of a terminated tableau. Then, there exists a valuation v induced by b such that for every formula F:

(i) If $1(F)$ is in b, $v(F) = 1$,
(ii) If $0(F)$ is in b, $v(F) = 0$.

Proof. The proof is by induction on the complexity of F (Definition 7).

(1) If F is a literal, $\circ A$, $\bullet A$, $\neg\circ A$, or $\neg\bullet A$, define $v(F) = s(F)$.

(2) $F = A \wedge B$.

(2.1) If $1(A \wedge B)$ is in b, since the tableau is terminated, **Rule 1** has been applied, therefore $1(A)$ is in b and $1(B)$ is in b. By inductive hypothesis, $v(A) = 1$ and $v(B) = 1$. We then define $v(A \wedge B) = 1$.

(2.2) If $0(A \wedge B)$ is in b, **Rule 2** was applied and the tableau bifurcated into two branches: b' and b''. In b', we have $0(A)$, and in b'', $0(B)$. Since b is an open branch of a terminated tableau, we have two (non-excluding) options: (i) $1(A)$ is not in b; (ii) $1(B)$ is not in b. In the case (i), b' is an open branch, and by inductive hypothesis, $v(A) = 0$. In the case (ii), b'' is an open branch, and by inductive hypothesis, $v(B) = 0$. In both cases define $v(A \wedge B) = 0$.

The valuation v defined in (2.1) and (2.2) clearly satisfies Definition 1.

(3) $F = \neg(A \wedge B)$.

(3.1) If $1(\neg(A \wedge B))$ is in b, since the tableau is terminated, **Rule 3** was applied and the tableau bifurcated into two branches: b' and b''. In b', we have $1(\neg A)$, and in b'' $1(\neg B)$. Since the b is an open branch of a terminated tableau, we have two (non-excluding) options: (i) $0(\neg A)$ is not in b, (ii) $0(\neg B)$ is not in b. If (i), b' is an open branch and, by inductive hypothesis, $v(\neg A) = 1$. Define $v(\neg(A \wedge B)) = 1$. If (ii), b'' is an open branch and, by inductive hypothesis, $v(\neg B) = 1$. Define $v(\neg(A \wedge B)) = 1$.

(3.2) If $0(\neg(A \wedge B))$ is in b, then, since the tableau is terminated, **Rule 4** was applied, therefore $0(\neg A)$ is in b and $0(\neg B)$ is in b. But since the tableau is open, $1(\neg A)$ is not in b and $1(\neg B)$ is not in b. Then, by inductive hypothesis, $v(\neg A) = 0$ and $v(\neg B) = 0$. Define $v(\neg(A \wedge B)) = 0$.

The valuation v defined in (3.1) and (3.2) satisfies Definition 1.

The cases (4) $F = A \vee B$ and (5) $F = \neg(A \vee B)$ are left to the reader.

(6) $F = \neg\neg A$.

If $1(\neg\neg A)$ is in b, by **Rule 9**, $1(A)$ is also in b. By inductive hypothesis, $v(A) = 1$, and we define $v(\neg\neg A) = 1$. By analogous reasoning if $0(\neg\neg A)$ in b, we define $v(\neg\neg A) = 0$.

The valuation v so defined satisfies Definition 1.

We have just shown that the valuation v defined above satisfies clauses (v1) to (v5) of Definition 1. It remains to be shown that v also satisfies clauses (v6) and (v7).

(7) $F = \circ A$.

(7.1) If $1(\circ A)$ is in b, then by item (1) above, $v(\circ A) = 1$ and $v(\bullet A) = 0$. As the tableau is terminated, **Rule 11** was applied and the tableau bifurcated into two branches: b' and b'' such that (i) $1(A)$ and $0(\neg A)$ occur in b', and (ii) $0(A)$ and $1(\neg A)$ occur in b''. If b' is open, then by inductive hypothesis $v(A) = 1$ and $v(\neg A) = 0$, and if b'' is open, then by inductive hypothesis $v(A) = 0$ and $v(\neg A) = 1$. So, v satisfies Definition 1 (clause (v6)).

(7.2) If $0(\circ A)$ is in b, then, as b is open, $1(\circ A)$ is not in b, and by item (1) above, $v(\circ A) = 0$ and $v(\bullet A) = 1$.

(8) $F = \bullet A$.

(8.1) If $1(\bullet A)$ is in b, then, as the tableau is terminated, **Rule 12** was applied and $0(\circ A)$ is in b. Since the tableau is open, $1(\circ A)$ is not in b. By item (1) above, $v(\bullet A) = 1$ and $v(\circ A) = 0$.

(8.2) If $0(\bullet A)$ is in b, then, as the tableau is terminated, $1(\circ A)$ is in b. By item (1) above, $v(\bullet A) = 1$ and $v(\circ A) = 0$.

(9) $F = \neg\circ A$.

If $1(\neg\circ A)$ is in b, $v(\neg\circ A) = 1$ by definition. If $1(\neg\circ A)$ is not in b, $v(\neg\circ A) = 0$ by definition. Analogous reasoning applies to $F = \neg\bullet A$.

Therefore, v as defined is a legitimate LET_F-valuation. □

Theorem 14. [Completeness]

If $\Gamma \vDash A$, then $\Gamma \vdash A$.

Proof. We prove the contrapositive: if $\Gamma \nvdash A$, then $\Gamma \nvDash A$. Suppose $\Gamma \nvdash A$. Thus there is a terminated LET_F-tableau with at least one open branch b such that $0(A)$ is in b and for every formula F from Γ, $1(F)$ is in b. By Lemma 13, there is a LET_F-valuation v induced by b such that if $1(F)$ is in b, $v(F) = 1$, and if $0(F)$ is in b, $v(F) = 0$. Therefore, there is a LET_F-valuation v such that $v(A) = 0$, and for every formula F from Γ, $v(F) = 1$. Therefore, $\Gamma \nvDash A$. □

Clearly, the tableau system for LET_F introduced here is equivalent to the natural deduction formulation presented in [19]. Indeed, Theorem 14 shows that the tableau system for LET_F is semantically characterized by the same valuation semantics (Definition 4.2) that characterizes the natural deduction rules for the version of LET_F introduced in [19].

4.3 Decidability

Definition 15. [Generalized subformula]

1. If B is a subformula of A (in the usual sense) and $A \neq B$, then B is an immediate subformula of A.
2. If B is an immediate subformula of A, then B is a generalized subformula of A.
3. $\neg A$ and $\neg B$ are generalized subformulas of both $\neg(A \wedge B)$ and $\neg(A \vee B)$.
4. $\neg A$ is a generalized subformula of $\circ A$.
5. $\circ A$ is a generalized subformula of $\bullet A$.
6. If C is a generalized subformula of B and B is an generalized subformula of A, then C is a generalized subformula of A.

As a consequence of the definition above, both A and $\neg A$ are generalized subformulas of $\circ A$ and $\bullet A$, since $\circ A$ is a generalized subformula of $\bullet A$. Besides, in view of the Definition 15, it is easy to see that if B is a generalized subformula of A, then $\mathcal{C}(B) < \mathcal{C}(A)$.

Theorem 16. [Decidability]

LET_F-tableaux provide a decision procedure for LET_F.

Proof. Clearly, every term occurring in a LET_F-tableau of $\Gamma \vdash A$ consists of signed formulas of $\Gamma \cup \{A\}$ (in the first node) and of signed generalized subformulas of $\Gamma \cup \{A\}$ (in the subsequent nodes), and each tableau rule yields generalized subformulas of the formula to which the rule is applied. Since the complexity of formulas occurring in the tableau is monotonically decreasing by applications of rules, all tableau branches are either closed or reach formulas of less complexity for which there is no rule to be applied, namely, a literal (with label 0 or 1), $\circ A$ (with label 0), $\neg\circ A$ or $\neg\bullet A$ (with label 0 or 1). Therefore LET_F-tableaux provide a decision procedure for LET_F. □

5 Some Examples of LET_F-Tableaux

In this section, we give some examples of tableaux that illustrate properties of LET_F.

Example 17. [Bottom particle]

A bottom particle can be defined in LET_F as $p \wedge \neg p \wedge \circ p$, and clearly, $\bot \vdash B$, for any B.

$(p \wedge \neg p) \wedge \circ p \vdash q$

1.	$1((p \wedge \neg p) \wedge \circ p))$		
2.	$0(q)$		
3.	$1(p \wedge \neg p)$		Rule 1 in 1
4.	$1(\circ p)$		Rule 1 in 1
5.	$1(p)$		Rule 1 in 3
6.	$1(\neg p)$		Rule 1 in 3
7.	$1(p)$	$0(p)$	Rule 11, 4
8.	$0(\neg p)$	$1(\neg p)$	Rule 11, 4
	$\otimes$ 6, 8	$\otimes$ 5, 7	

Every LFI has a bottom particle, since in every LFI a bottom particle can be defined as above. This is because the principle of gentle explosion (item (3) page 326) is an essential feature of LFIs.

Example 18. [Recovering modus ponens]
As expected, disjunctive syllogism does not hold in LET_F, and so modus ponens, since the natural way of defining $A \to B$ in LET_F is as $\neg A \vee B$.

$p, \neg p \vee q \nvdash q$

1.	$1(p)$		
2.	$1(\neg p \vee q)$		
3.	$0(q)$		
4.	$1(\neg p)$	$1(q)$	Rule 5 in 2
		$\otimes$ 3, 4	

The open branch gives a counterexample: $v(p) = v(\neg p) = 1$ and $v(q) = 0$. For classical p modus ponens is recovered, as we see below [cf. 19, Fact 32].

$\circ p, p, \neg p \vee q \vdash q$

1.		$1(\circ p)$		
2.		$1(p)$		
3.		$1(\neg p \vee q)$		
4.		$0(q)$		
5.	$1(\neg p)$		$1(q)$	Rule 5 in 3
6.	$1(p)$	$0(p)$	$\otimes$ 4, 5	Rule 11 in 1
7.	$0(\neg p)$	$1(\neg p)$		Rule 11 in 1
	$\otimes$ 5, 7	$\otimes$ 2, 6		

Example 19. [Proofs by cases]

Excluded middle does not hold in LET_F, so the usual form of proof by cases does not obtain. But LET_F allows other forms of proof by cases, e.g., from $\vdash \circ p \vee \bullet p$ and $\vdash p \vee \neg p \vee \bullet p$.

$\vdash \circ p \vee \bullet p$

1.	$0(\circ p \vee \bullet p)$	
2.	$0(\circ p)$	Rule 6 in 1
3.	$0(\bullet p)$	Rule 6 in 1
4.	$1(\circ p)$	Rule 13 in 3
	$\otimes$ 2, 4	

The operators $\bullet$ and $\circ$ work as if $\bullet A$ were a classical negation of $\circ A$, and vice-versa. Indeed, $\circ A, \bullet A \vdash B$ prohibits that $\circ A$ and $\bullet A$ hold together, on pain of triviality. Note, however, that $\circ A \vee \neg\circ A$ is not valid, as we see below.

$\nvdash \circ p \vee \neg\circ p$

1.	$0(\circ p \vee \neg\circ p)$	
2.	$0(\circ p)$	Rule 6 in 1
3.	$0(\neg\circ p)$	Rule 6 in 1

Formulas $\circ p$ and $\neg\circ p$ are independent of each other, as we have seen in the quasi-matrice of Example 3.

The Example 19 above indicates that $\neg\circ p$ and $\bullet p$ are not equivalent, as we see below.

Example 20.

$\neg\circ p \nvdash \bullet p$

1.	$1(\neg\circ p)$		
2.	$0(\bullet p)$		
3.	$1(\circ p)$		Rule 13 in 2
4.	$1(p)$	$0(p)$	Rule 11 in 3
5.	$0(\neg p)$	$1(\neg p)$	Rule 11 in 3

The proof of $\bullet p \nvdash \neg\circ p$ is left to the reader. In LET_F, negation of classicality does not entail classicality because the negation $\neg$ is still a weak negation. The same applies to $\circ p$ and $\neg\bullet p$, which are not equivalent. As far as we know, classical negation cannot be defined in LET_F [cf. 19, footnote 15]).

Example 21. [On 'quasi-negations' in LET_F]

Two unary connectives that are (in some sense) negations can be defined in LET_F.

(1) $\sim A := \circ A \wedge \neg A$,
(2) $\approx A := \bullet A \vee \neg A$.[4]

We have that explosion holds for $\sim$ and excluded middle for $\approx$, i.e.,

(3) $A, \sim A \vdash B$,
(4) $\vdash A \vee \approx A$,

as well as the following dual inferences,

(5) $\sim A \vdash \neg A$,
(6) $\neg A \vdash \approx A$.

On the other hand, neither double negation nor de Morgan hold for $\sim$ and $\approx$ in LET_F. For this reason, we think these connectives should rather be called 'quasi-negations'. We prove below $p \nvdash \approx\approx p$ and $\sim(p \vee q) \nvdash \sim p \wedge \sim q$, and leave to the reader the other invalid inferences with $\sim$ and $\approx$.

[4]The connectives $\sim$ and $\approx$ are called, respectively, supplementing and complementing negation [cf. 19, Def. 33].

$p \nvdash \bullet(\bullet p \vee \neg p) \vee \neg(\bullet p \vee \neg p)$

1.	$1(p)$	
2.	$0(\bullet(\bullet p \vee \neg p) \vee \neg(\bullet p \vee \neg p))$	
3.	$0(\bullet(\bullet p \vee \neg p))$	Rule 6 in 2
4.	$0(\neg(\bullet p \vee \neg p))$	Rule 6 in 2
5.	$1(\circ(\bullet p \vee \neg p))$	Rule 13 in 3
6.	$0(\neg\bullet p)$; $0(\neg\neg p)$	Rule 6 in 4
7.	$\vdots$	
8.	$1(\bullet p \vee \neg p)$; $0(\bullet p \vee \neg p)$	Rule 11 in 5
9.	$0(\neg(\bullet p \vee \neg p))$; $1(\neg(\bullet p \vee \neg p))$	Rule 11 in 5
10.	$\vdots$	
11.	$1(\bullet p)$; $1(\neg p)$	Rule 5 in 8
12.	$\vdots$	
13.	$0(\neg\bullet p)$; $0(\neg\neg p)$	Rule 8 in 9
14.	$\vdots$	

$\circ(p \vee q) \wedge \neg(p \vee q) \nvdash (\circ p \wedge \neg p) \wedge (\circ q \wedge \neg q)$

1.	$1(\circ(p \vee q) \wedge \neg(p \vee q))$				
2.	$0((\circ p \wedge \neg p) \wedge (\circ q \wedge \neg q))$				
3.	$1(\circ(p \vee q))$				Rule 1 in 1
4.	$1(\neg(p \vee q))$				Rule 1 in 1
5.	$1(\neg p)$				Rule 7 in 4
6.	$1(\neg q)$				Rule 7 in 4
7.	$1(p \vee q)$		$0(p \vee q)$		Rule 11 in 3
8.	$0(\neg(p \vee q))$		$1(\neg(p \vee q))$		Rule 11 in 3
9.	$0(\neg p)$	$0(\neg q)$			Rule 8 in 8
10.	$\otimes$ 5	$\otimes$ 6	$0(p)$		Rule 6 in 7
11.			$0(q)$		Rule 6 in 7
12.			$1(\neg p)$		Rule 7 in 8
13.			$1(\neg q)$		Rule 7 in 8
14.			$0(\circ p \wedge \neg p)$	$0(\circ q \wedge \neg q)$	Rule 2 in 2
15.			$0(\circ p)$; $0(\neg p)$ $\otimes$ 12	$0(\circ q)$; $0(\neg q)$ $\otimes$ 13	Rule 2 in 14

A central point of the intended interpretation of *LET*s is that there may be scenarios in which there is evidence for exactly one among A and $\neg A$, but such evidence is still not conclusive. In this case, $A \vee \neg A$ holds, as well $\neg(A \wedge \neg A)$, but $\circ A$ does not hold. Indeed, in a scenario such that A holds but $\neg A$ does not hold, the evidence for A may be non-conclusive, and so A cannot be taken as true and is not subjected to classical logic. This idea is expressed in the semantics by means of clause (v6), which states only a necessary condition for $\circ A$. The same idea could be expressed equivalently, with $\bullet$, by the contrapositive of clause (v6) – indeed, when both A and $\neg A$ hold, as well as when neither holds, $\bullet A$ holds. The behavior of $\circ$ and $\bullet$ is illustrated by Example 22 below.

Example 22. [On the behavior of $\circ$ and $\bullet$]

$p, \neg p \vdash \bullet p$

1.	$1(p)$		
2.	$1(\neg p)$		
3.	$0(\bullet p)$		
4.	$1(\circ p)$		Rule 13 in 3
5.	$1(p)$	$0(p)$	Rule 11 in 4
6.	$0(\neg p)$	$1(\neg p)$	Rule 11 in 4
	$\otimes$ 2, 6	$\otimes$ 1, 5	

$\bullet p \nvdash p \wedge \neg p$

1.	$1(\bullet p)$		
2.	$0(p \wedge \neg p)$		
3.	$0(\circ p)$		Rule 12 in 1
4.	$0(p)$	$0(\neg p)$	Rule 2 in 2

Proofs of $\circ p \vdash p \vee \neg p$ and $p \vee \neg p \nvdash \circ p$ are left to the reader.

Example 23. [Propagation of classicality]

The operator $\circ$ does not propagate over more complex formulas, that is, $\circ A, \circ B$ does not imply $\circ\neg A$, $\circ(A \wedge B)$, or $\circ(A \vee B)$. We illustrate this fact showing that $\circ p, \circ q \nvdash \circ(p \wedge q)$.

$\circ p, \circ q \nvdash \circ(p \wedge q)$

1.	$1(\circ p)$				
2.	$1(\circ q)$				
3.	$0(\circ(p \wedge q))$				
4.	$1(\neg p)$		$0(\neg p)$		Rule 11 in 1
5.	$0(p)$		$1(p)$		Rule 11 in 1
6.	$1(\neg q)$	$0(\neg q)$	$1(\neg q)$	$0(\neg q)$	Rule 11 in 2
7.	$0(q)$	$1(q)$	$0(q)$	$1(q)$	Rule 11 in 2

Notice, however, that the set $\{\circ p, \circ q, p \wedge q, \neg(p \wedge q)\}$ is not satisfiable. Indeed, although in LET_F $\circ$ does not propagate over $\neg$, $\wedge$, and $\vee$, if $\circ p_1, ..., \circ p_n$ hold, then all formulas formed with $\{p_1, ..., p_n\}$ over $\{\neg, \wedge, \vee\}$ behave classically, as has been shown in [19, Fact 31].

6 Final Remarks

Analytic tableaux constitute a decision procedure for LET_F that, we think, is at least more elegant than quasi-matrices. However, besides issues of elegance, it is reasonable to conjecture that LET_F-tableaux are in fact more efficient than the quasi-matrices.

Although it is well known that analytic tableaux (at least for standard logics) may require super-exponential time and are subject to other issues in complexity [9, 15], such intractability obstacles are not immediately generalizable for non-classical cases. Moreover, in some cases modifications of the tableau method can render certain classes tractable. Non-deterministic matrices, like the ones yielded by the semantics of LET_F presented in [19], are certainly less efficient in space than classical truth-tables, but at first sight this does not occur with respect to LET_F-tableaux when compared to classical tableaux. In the case of LET_F, the examples of Section 5 suggest that, at least locally, for some classes of formulas, tableaux are indeed better than quasi-matrices. For this reason, whether or not the results of [9, 15] apply to LET_F is a question that deserves to be further investigated.

Considering its potential applications in automated reasoning and artificial intelligence, LET_F deserves in-depth investigations. This paper is one more step in this direction.

Acknowledgments

Walter Carnielli was supported by the grant 307376/2018-4, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil). Abilio Rodrigues was supported by the grants 311911/2018-8, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil), and APQ-02093-21, Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG, Brazil).

References

[1] H. Antunes, W. Carnielli, A. Kapsner, and A. Rodrigues. Kripke-style models for logics of evidence and truth. *Axioms*, 9(3):100, 2020. DOI: 10.3390/axioms9030100.

[2] N. D. Belnap. How a computer should think. In G. Ryle (ed.), *Contemporary Aspects of Philosophy*, pp. 30–55. Oriel, 1977. Reprinted in [16, 35–53].

[3] A. Bergstra and A. Ponse. Process algebra with four-valued logic. *Journal of Applied Non-Classical Logics*, 10:27–53, 2000. DOI: 10.1080/11663081.2000.10510987.

[4] J. Bueno-Soler, W. Carnielli, M. Coniglio, and A. Rodrigues. Applications of logics of evidence and truth to automated reasoning. Unpublished.

[5] W. Carnielli and J. Marcos. Tableau systems for logics of formal inconsistency. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pp. 848–852. CSREA, 2001. URL: `https://www.ijcai.org/Proceedings/2001-2`.

[6] W. Carnielli and A. Rodrigues. An epistemic approach to paraconsistency: a logic of evidence and truth. *Synthese*, 196:3789–3813, 2017. DOI: 10.1007/s11229-017-1621-7.

[7] W. Carnielli, M. Coniglio, and J. Marcos. Logics of formal inconsistency. In D. M. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, volume 14, pp. 1–93. Springer, 2007.

[8] W. Carnielli, M. Coniglio, and A. Rodrigues. Recovery operators, paraconsistency and duality. *Logic Journal of the IGPL*, 28(5):624—-656, 2019. DOI: 10.1093/jigpal/jzy054.

[9] M. D'Agostino. Are tableaux an improvement on truth-tables? *Journal of Logic, Language, and Information*, 1:235–252, 1992. DOI: 10.1007/BF00156916.

[10] J. M. Dunn. Intuitive semantics for first-degree entailments and 'coupled trees'. *Philosophical Studies*, 29(3):149–168, 1976. Reprinted in [16, pp. 21–53].

[11] M. L. Ginsberg. Multi-valued logics: A uniform approach to reasoning in artificial intelligence. *Computer Intelligence*, 4:256–316, 1988. DOI: 10.1111/j.1467-8640.1988.tb00280.x.

[12] A. Loparic. A semantical study of some propositional calculi. *The Journal of Non-Classical Logic*, 3(1):73–95, 1986.

[13] A. Loparic. Valuation semantics for intuitionistic propositional calculus and some of its subcalculi. *Principia*, 14(1):125–133, 2010.

[14] J. Marcos. Nearly every normal modal logic is paranormal. *Logique et Analyse*, Nouvelle série, 48(189–192):279–300, 2005. URL: http://jstor.org/stable/44084807.

[15] N. V. Murray and E. Rosenthal. On the computational intractability of analytic tableau methods. *Logic Journal of the IGPL*, 2(2):205–228, 1994. DOI: 10.1093/jigpal/2.2.205.

[16] H. Omori and H. Wansing (eds.). *New Essays on Belnap-Dunn Logic*. Number 418 in Synthese Library Studies in Epistemology, Logic, Methodology, and Philosophy of Science. Springer, 2019.

[17] P. F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38(3):319–351, 1989. DOI: 10.1016/0004-3702(89)90036-2.

[18] A. Rodrigues and W. Carnielli. On Barrio, Lo Guercio, and Szmuc on logics of evidence and truth. Forthcoming.

[19] A. Rodrigues, J. Bueno-Soler, and W. Carnielli. Measuring evidence: a probabilistic approach to an extension of Belnap-Dunn logic. *Synthese*, 198:5451—-5480, 2021. DOI: 10.1007/s11229-020-02571-w.

[20] U. Straccia. A sequent calculus for reasoning in four-valued description logics. In D. Galmiche (ed.), *Automated Reasoning with Analytic Tableaux and Related Methods*, pp. 343–357. Springer, 1997.

Walter Carnielli
CLE-University of Campinas (UNICAMP)
Rua Sérgio Buarque de Holanda, 251 — Cidade Universitária,
13083-859 Campinas, SP, Brazil
E-mail: walterac@unicamp.br

Lorenzzo Frade
Federal University of Minas Gerais (UFMG)
Av. Antônio Carlos, 6627, CEP 31270-901, Belo Horizonte, MG, Brazil
E-mail: lorenzzo_frade@hotmail.com

Abilio Rodrigues
Federal University of Minas Gerais (UFMG)
Av. Antônio Carlos, 6627, CEP 31270-901, Belo Horizonte, MG, Brazil
E-mail: abilio.rodrigues@gmail.com