Analytic classic-like tableaux for non-deterministic many-valued logics

(Extended Abstract)

Filipa da Silva Montez

Supervisor: Prof. Carlos Manuel Costa Lourenço Caleiro

Instituto Superior Técnico - Universidade de Lisboa

June 2018

Abstract

This work focuses on the development of a proof procedure for non-deterministic many-valued logics. The intrinsic bivalence that underlies many-valued logics can be used to our advantage by representing such semantics using only two logical truth values. As such when the linguistic resources are expressive enough to allow such representation, an equivalent bivalent semantics can be found, along with a constructive procedure. A tableaux proof-system can be extracted from the bivalent semantics obtained, together with a strategy to ensure termination of the method. This system preserves some classic-like features as the labels used in rules adopt exactly two signs, and is guaranteed to be sound and complete, as well as to satisfy a general notion of analiticity. Finally, this work also reports on the implementation of an application for extracting and employing the aforementioned classic-like proof systems to such classes of semantics.

1 Introduction

The truth-functional nature of many-valued semantics, meaning that the truth of a compound sentence is determined by the truth values of its component sentences, makes them particularly attractive from a linguistic point of view (see [8]). They have been used to deal with assumptions in linguistics and model natural language fenomena, as well as in formalizing languages with their own truth predicates. Furthermore in artificial intelligence many-valued logics can be used in the automation of data and knowledge mining. However many-valued logics do not necessarily capture real-world information where information may be incomplete, uncertain or inconsistent (see [3]). Therefore to capture the non-deterministic behaviour from several sources available, the concept can be applied to the evaluation of formulas, giving rise to non-deterministic semantics. Furthermore there are several logics which can not be specified by means of a finite deterministic many-valued characteristic matrix, but can be specified by finite non deterministic many-valued ones, making them decidable (see [3], [1]). This can be useful in modeling linguistic ambiguities, the behaviour of circuits which may deviate from the expected behavior, model computations with unknown functions or verification with unknown evaluation models (see [2]).

The bivalence underlying many-valued logics can be used to represent such systems by means of a semantics with only two logical values, that is, a bivalent semantics (see [6]). Furthermore such reduction can be performed in a constructive way for deterministic many-valued logics, as shown in [4]. A method for extracting a classic-like tableaux from deterministic many-valued semantics is presented in [5]. In this method, the classic-like 2-signed tableau system is constructively extracted from the corresponding finite-valued semantics while guaranteeing a general concept of analiticity. Our aim is to take a similar path to develop a tableau proof system for many-valued non-deterministic logics. Sections 2 and 3 focus on the development of such procedure, while Section 4 is dedicated to a web application implementing the method developed throughout the previous Sections.

2 Bivalent semantics for non-deterministic many-valued logics

2.1 Non-deterministic many-valued logics

Let $\mathcal{A} = \{p_0, p_1, ...\}$ be a denumerable set of atomic variables representing an alphabet and $\Sigma = \{\odot_0, ..., \odot_k\}$ a finite collection of primitive connectives. For each connective in Σ its arity can be defined as $ar : \Sigma \to \mathbb{N}_0$. The set Σ_k represents the collection of all k-ary connectives, with 0-ary connectives being logical constants. The set $\mathcal{S}$ of formulas is the carrier of the Σ-algebra $\mathbb{S}$ generated by $\mathcal{A}$.

Definition 1. A non-deterministic many-valued logic $\mathcal{L}$ can be defined by a non-deterministic matrix $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:

- $\mathcal{V} = \{v_0, ..., v_{n-1}\}$ is a non-empty set comprising the truth-values of the logic;
- the non-empty partition $\mathcal{D} \subset \mathcal{V}$ represents the set of designated values, and $\mathcal{U} = \mathcal{V} \backslash \mathcal{D}$ contains the undesignated values of the logic;
- for every k-ary connective of $\mathcal{L}$, $\mathcal{O}$ includes its interpretation $\widehat{\odot} : \mathcal{V}^k \to 2^{\mathcal{V}} \backslash \{\emptyset\}$.

Definition 2. A valuation is a function $w : \mathcal{S} \to \mathcal{V}$, that is, an extension of any assignment $\rho : \mathcal{A} \to \mathcal{V}$ to the set of all formulas such that for every k-ary connective $\odot$ of the logic and formulas $\varphi_1, ..., \varphi_k$ satisfying:

$$w(\odot(\varphi_1, ..., \varphi_k)) \in \widehat{\odot}(w(\varphi_1), ..., w(\varphi_k)). \tag{1}$$

Note that according to Definition 1 a non-deterministic many-valued logic has at least two truth values, one designated and one undesignated. Without loss of generality let those values be v_{n-1} and v_0 respectively. In the course of this work $v_0 \in \mathcal{U}$ will be referred to as F and $v_{n-1} \in \mathcal{D}$ as T. The set $\mathcal{V}_2$ will be used to denote $\{T, F\}$. Furthermore, as an abuse of notation, we will use T^C to refer to F and F^C to refer to T.

Definition 3. A valuation w is a model of a formula φ if $w(\varphi) \in \mathcal{D}$.

A semantics for the logic is obtained by taking Sem as the set of valuations of the logic, and defining the notion of entailment $\models_{\mathsf{Sem}}$ associated to the semantics as $(\Gamma \models_{\mathsf{Sem}} \alpha)$ iff $(w[\Gamma] \subseteq \mathcal{D} \implies w(\alpha) \in \mathcal{D})$ for any arbitrary $\Gamma \cup \{\alpha\} \subseteq \mathcal{S}$ and valuation $w \in \mathsf{Sem}$. The logic $\mathcal{L}$ defined by Nmatrix $\langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$ is said to be n-valued if $|\mathcal{V}| = n$. The n-valued logic with entailment relation $\models_{\mathsf{Sem}}$ is genuinely n-valued if there exists no $m < n$ such that $\models_{\mathsf{Sem}}$ can be obtained by a semantics with m truth values.

Definition 4. A formula $\varphi \in \mathcal{S}$ is said to be valid if $w(\varphi) \subseteq \mathcal{D}$ for all valuations $w \in \mathsf{Sem}$. In this case it is written $\models \varphi$, and φ is said to be a theorem of the logic.

A semantic is called bivalent when $\mathcal{V} = \mathcal{V}_2$ and $\mathcal{D} = \{T\}$. In this case the corresponding valuations are called bivaluations. Any multivalued logic can be characterized by a classic-like bivalent semantics. In order to do so let $t : \mathcal{V} \to \mathcal{V}_2$ be the total map such that:

$$t(v) = \begin{cases} T & \text{if } v \in \mathcal{D} \\ F & \text{if } v \in \mathcal{U} \end{cases}$$

Definition 5. Let $\mathcal{L}$ be an non-deterministic many-valued logic with semantics Sem. Then the corresponding bivalent semantics is the set $\mathsf{Sem}_2 = \{b_w = t \circ w : w \in \mathsf{Sem}\}$.

Proposition 1. Semantics Sem and Sem_2 characterize the same logic.

Proof. In the following it is proven that $\Gamma \models_{\mathsf{Sem}} = \models_{\mathsf{Sem}_2}$. Let $\varphi \in \mathcal{S}$ be an arbitrary formula.

$$\begin{array}{llll} \Gamma \models_{\mathsf{Sem}} \varphi & \text{iff} & w(\Gamma) \subseteq D \text{ implies } w(\varphi) \subseteq D & \forall w \in \mathsf{Sem} \\ & \text{iff} & t(w(\Gamma)) = \{T\} \text{ implies } t(w(\varphi)) = \{T\} & \forall w \in \mathsf{Sem} \\ & \text{iff} & b_w(\Gamma) = T \text{ implies } b_w(\varphi) = T & \forall b_w \in \mathsf{Sem}_2 \\ & \text{iff } \Gamma & \models_{\mathsf{Sem}_2} \varphi & \end{array}$$

□

The semantics produced is clearly two-valued. But for genuinely n-valued logics where $n > 2$ the principle of truth-functionality is lost. However a generalised notion of compositionality still holds, in which the value of a formula is to be (uniquely) determined from the values of separators applied to its immediate subformulas. Though describing the logic as a way of a bivalent semantics is useful, one still needs to find a way of recovering information about the original algebraic values, and distinguishing each pair of values using the linguistic resources of the logic.

Definition 6. The truth values $v_i, v_j \in \mathcal{V}$ are said to be separated, written $v_i \# v_j$, if $t(v_i) \neq t(v_j)$. Given sets of truth values A and B, they are said to be separated, written $A\#B$, if one set only has designated values and the other undesignated ones, meaning $t(A) \cap t(B) = \emptyset$. The values v_i and v_j are distinguishable if there exists a formula $\theta^{ij}(p) \in \mathcal{S}$, called separator, that separates v_i and v_j, that is

$$\widehat{\theta^{ij}}(v_i)\#\widehat{\theta^{ij}}(v_j) \equiv t(\widehat{\theta^{ij}}(v_i)) \cap t(\widehat{\theta^{ij}}(v_j)) = \emptyset. \tag{2}$$

A logic is said to be separable if its logical values $\langle v_i, v_j \rangle \in \mathcal{D}^2 \cup \mathcal{U}^2$ are pairwise distinguishable. In this case the language of the logic in question is said to be *sufficiently expressive*. When a logic is separable, separators for all the possible pairs of truth values are collected without repetition into a finite sequence $\bar{\theta} = \langle \theta_r \rangle_{r=0}^{s}$, with θ_0 being the identity mapping. To the construction of our tableau system we will only allow separators to be unary connectives of the logic.

Definition 7. Given a separable logic $\mathcal{L}$ with separating sequence $\bar{\theta} = \langle \theta_r \rangle_{r=0}^{s}$, the general binary print of some truth value $v \in \mathcal{V}$ is defined as the sequence $\bar{\theta}(v) = \langle t(\widehat{\theta_r}(v))] \rangle_{r=0}^{s}$. A definite binary print of logical value $v \in \mathcal{V}$ is defined as $\bar{\xi} = \langle \xi_r \rangle_{r=0}^{s}$, where $\xi_r \in t(\widehat{\theta_r}(v))$ for every $0 \leq r \leq s$.

Let $\Xi(v)$ represent the set of all definite binary prints of a given logical value $v \in \mathcal{V}$. Note that not all definite binary prints $\xi \in {\mathcal{V}_2}^{s+1}$ correspond to some truth value $v \in \mathcal{V}$.

Definition 8. A definite binary print $\bar{X}$ is said to be unobtainable if there is no $v \in \mathcal{V}$ such that $\bar{X} \in \Xi(v)$. Otherwise $\bar{X}$ is called obtainable.

Note truth values may have more than one definite binary print associated. However by construction no two values share the same general or definite binary prints. This reveals an unambiguous correspondence between the prints and the truth values of a logic, and can be expressed in the following definition.

Definition 9. The surjective map $\pi : \Xi[\mathcal{V}] \to \mathcal{V}$ attributes to each definite binary print the unique truth value z for which $\bar{X} \in \Xi(z)$.

Example 1. *Consider the connectives* $\Sigma = \{\blacklozenge, *, -, \bullet\}$*, and the logic* $\mathcal{L}$ *with the logical values* $\mathcal{V} = \{1, 2, 3, 4\}$. *Take* $\mathcal{D} = \{1, 2, 3\}$ *as designated values and consider the interpretation of the logic operators is defined in Figure 1.*

$\widehat{\blacklozenge}$	1	2	3	4
1	{1}	{1}	{1}	{1}
2	{1}	{1,2}	{2}	{2}
3	{1}	{2}	{2,3}	{3}
4	{1}	{2}	{3}	{3,4}

$\widehat{*}$	1	2	3	4
1	{1}	{1}	{1}	{1}
2	{1}	{2,4}	{2,3,4}	{4}
3	{1}	{2,3,4}	{3,4}	{4}
4	{4}	{4}	{4}	{4}

$\widehat{-}$	
1	{1}
2	{1}
3	{4}
4	{1}

$\widehat{\bullet}$	
1	{1}
2	{4}
3	{4}
4	{3,4}

Figure 1: Interpretation of non-deterministic operators $\blacklozenge$, $*$, $-$ and $\bullet$.

One can consider separators $\theta_1(p) = \bullet(p)$ *and* $\theta_2(p) = -(p)$ *and build the separating sequence* $\theta = \langle \theta_0, \theta_1, \theta_2 \rangle$ *for the logic. The general binary prints associated with each of the logical values are* $\bar{\theta}(1) = \langle \{T\}, \{T\}, \{T\} \rangle$, $\bar{\theta}(2) = \langle \{T\}, \{F\}, \{T\} \rangle$, $\bar{\theta}(3) = \langle \{T\}, \{F\}, \{F\} \rangle$ *and* $\bar{\theta}(4) = \langle \{F\}, \{T, F\}, \{T\} \rangle$. *As for the definite binary prints of values, for example* $\Xi(1) = \{\langle T, T, T \rangle\}$ *and* $\Xi(4) = \{\langle F, T, T \rangle, \langle F, F, T \rangle\}$. *Note the binary print* $\langle T, T, T \rangle$ *is obtainable as it is an element of* $\Xi(1)$. *On the other hand* $\langle F, F, F \rangle$ *is an unobtainable binary print.*

2.2 Constructive characterization of bivalent semantics

In what follows a description of classic-like semantic Sem_2 is provided. For that, let $\|$ represent disjunction, & represent conjunction, and $\Longrightarrow$ represent implication. Also, in this metalanguage let $\perp$ represent absurd, and consider all statements to have their expected classical interpretation.

Definition 10. A labelled formula is an expression of the form $X : \varphi$, where $X \in \mathcal{V}_2$ and φ is a formula of the logic.

Definition 11. A bivaluation b satisfies $X : \varphi$ if $b(\varphi) = X$. A valuation, on the other hand is said to satisfy a labelled formula $X : \varphi$ whenever $b_w(\varphi) = t(w(\varphi)) = X$.

At any given moment considering $w \in \mathsf{Sem}$ a formula φ has associated a certain general binary print $\bar{X} = \langle X_r \rangle_{r=0}^{s}$ that leads to the expression

$$X_0 : \varphi \quad \& \quad X_1 : \theta_1(\varphi) \quad \& \quad \dots \quad \& \quad X_s : \theta_s(\varphi) \tag{V($\varphi;\bar{X}$)}$$

Note that $V(\varphi; \bar{\theta}(z))$ is satisfied by valuation w precisely when $w(\varphi) = z$. Naturally, this can be generalized to multiple formulas $\varphi_1, ..., \varphi_k$ and their corresponding binary prints $\bar{X}_1, ..., \bar{X}_k$:

$$V(\varphi_1; \bar{X}_1) \quad \& \quad \dots \quad \& \quad V(\varphi_k; \bar{X}_k) \tag{V($\varphi_1,\dots,\varphi_k;\bar{X}_1,...,\bar{X}_k$)}$$

A definite binary print is a sequence of $s+1$ elements in $\mathcal{V}_2$, thus $\{F, T\}^{s+1}$ comprises all the possible (s+1)-long definite binary prints. Then $\{F, T\}^{s+1} \setminus \Xi[\mathcal{V}]$ represents the definite binary prints that do not correspond to any logical value $v \in \mathcal{V}$, that is, all the unobtainable definite binary prints of the logic. This means that no formula leads to such prints and therefore for each $\bar{X} \in \{F, T\}^{s+1} \setminus \Xi[\mathcal{V}]$, and arbitrary formula φ follows that:

$$V(\varphi; \bar{X}) \Longrightarrow \perp. \tag{$U\bar{X}$}$$

Given a separating formula θ_r and $X \in \{T, F\}$ we now look at the k-long sequence of truth values the formulas $\varphi_1, \dots, \varphi_k$ may be assigned, such that the formula $\theta_r(\odot(\varphi_1, \dots, \varphi_k))$ can take the bivalent value X:

$$R_X^{\theta_r \odot} = \{\bar{v} \in \mathcal{V}^k : X \in t(\widehat{\theta_r}(\widehat{\odot}(\bar{v})))\}. \tag{3}$$

Now for each $\bar{v} = (v_1, \dots, v_k) \in R_X^{\theta_r \odot}$, we define the set $\Delta(\bar{v}) = \Xi(v_1) \times \dots \times \Xi(v_k)$, which contains all possible k-long tuples of definite binary prints such that $\pi(\Delta_{ij}) = v_j$ for each binary print Δ_{ij}. This way information obtained on $\theta_r(\odot(\varphi_1, \dots, \varphi_2))$ provides information on the subformulas of $\odot(\varphi_1, \dots, \varphi_2)$, and we arrive to the following statement for each $0 \le r \le s$, $\odot \in \Sigma$, and $X \in \{T, F\}$:

$$X : \theta_r(\odot(\varphi_1, \dots \varphi_k)) \Rightarrow \|_{\bar{v} \in R_X^{\theta_r \odot}} (\|_{\bar{\delta} \in \Delta(\bar{v})} V(\varphi_1, \dots, \varphi_k; \bar{\delta})). \tag{B$_X^{\theta_r \odot}$}$$

When $r = 0$ then $B_X^{\theta_r \odot}$ is said to be a non θ-statement, otherwise is called a θ-statement. Note that it can happen that $R_X^{\theta_r \odot} = \emptyset$, in which case the statement $B_X^{\theta_r \odot}$ is left with absurd in its right-hand side. Oppositely it may also be the case that $R_X^{\theta_r \odot} = \mathcal{V}$, and no restriction is imposed on the logical values the immediate subformulas of $\odot(\varphi_1, \dots, \varphi_k)$ can take.

The B-statements generated are typically not optimal and in general there are several simplifications that should be made in order to reduce the number of statements and their complexity. One obvious case is when the statement $B_X^{\theta_r \odot}$ is tautological, that is, when $R_X^{\theta_r \odot} = \mathcal{V}$. Such rule might just be omitted. Focus now on a rule $B_X^{\theta_r \odot}$, $\odot \in \Sigma_k$, whose right side is a disjunction of n clauses of the form $V(\varphi_1, \dots, \varphi_k; \bar{X}_1, \dots, \bar{X}_k)$. To obtain a simplified set of statements, one can reduce the right-hand side to its minimal equivalent boolean expression, by logic minimization procedures, while maintaining right-hand side in disjunctive normal form and keeping new labelled formulas from appearing. Because the new formula is equivalent to the old formula the validity of the method is not compromised.

In order to reach our desired characterization of Sem_2 one last kind of statements are going to be introduced, and to do so also a new concept of obtainability. Note that unlike the previously introduced concepts, this type of statements were not present in the deterministic procedure, and are introduced specifically to deal with non determinism.

Definition 12. Given $\odot \in \Sigma_k$ and $\bar{v} \in \mathcal{V}^k$, an obtainable definite binary print $\bar{X}$ is called $(\odot, \bar{v})$-unobtainable if:

- $\bar{X} \notin \Xi[\widehat{\odot}(\bar{v})]$
- for all $r \in \{0, \dots, s\}$ there is some $\xi \in \Xi[\widehat{\odot}(\bar{v})]$ such that $\xi_r = X_r$.

Let $\odot \in \Sigma_k$, $\varphi = \odot(\varphi_1, \dots, \varphi_k)$, $\bar{v} \in \mathcal{V}^k$, $\delta = \langle \bar{X}^{v_i} \rangle_{i=0}^k$ and $\bar{X}$ be a definite binary print. Assume bivaluation $b \in \mathsf{Sem}_2$ satisfies $V(\varphi_1, \dots, \varphi_k; \bar{X}^{v_1}, \dots, \bar{X}^{v_k})$. This provides us with information about the different values $b(\varphi)$ could take. Obviously $V(\varphi; \bar{X})$ can only be satisfied if $\pi(\bar{X}) \in \widehat{\odot}(v_1, \dots, v_k)$ or equivalently $\bar{X} \in \Xi[\widehat{\odot}(\bar{v})]$. When there is $0 \leq r \leq s$ such that $X_r \notin t(\widehat{\theta_r}(\widehat{\odot}(\bar{v})))$, then statement $B^{\theta_r \odot}_{X_r}$ will reflect the semantic impossibility $w_b(\varphi_i) \neq \bar{v}_i$ for some $1 \leq i \leq k$ and a contradiction will be reached. However given the non-determinism associated to the semantic, $\widehat{\odot}(\bar{v})$ may have more than one value. Thus there are particular cases when even though $\bar{X} \notin \Xi[\widehat{\odot}(\bar{v})]$, it is the case that $X_r \in t(\widehat{\theta_r}(\widehat{\odot}(\bar{v})))$ for each $0 \leq r \leq s$ and so the application of each rule $B^{\theta_r \odot}_{X_r}$ is not enough to reflect the semantic impossibility of $b(\varphi_i) = v_i$ for $1 \leq i \leq k$. This inconsistency is the motivation for introducing of the following statements for $(\odot, \bar{v})$-unobtainable binary prints $\bar{X}$:

$$V(\varphi_1, \dots, \varphi_k; \bar{X}^{v_1}, \dots, \bar{X}^{v_k}) \ \& \ V(\varphi; \bar{X}) \to \bot. \qquad (\mathrm{N}^{\odot \bar{X}}_{\delta})$$

Definition 13. Let $\mathcal{B}(\mathcal{L}, \bar{\theta})$ be the set of bivalent statements associated to a separable non-deterministic finite-valued logic $\mathcal{L}$ with separating sequence $\bar{\theta} = \langle \theta_r \rangle_{r=0}^s$, which consists of all instances of:

- $U\bar{X}$, for each unobtainable definite binary print $\bar{X}$, (U-Statements)
- $B^{\theta_r \odot}_{X}$, for each $X \in \{T, F\}$, $0 \leq r \leq s$ and $\odot \in \Sigma$, and (B-Statements)
- $N^{\odot \bar{X}}_{\delta}$, for each $\odot \in \Sigma_k$, $\bar{v} \in (\mathcal{V})^k$, $\delta \in \Delta(\bar{v})$ and each $(\odot, \bar{v})$-unobtainable binary print $\bar{X}$. (N-Statements)

Proposition 2. Sem_2 is the set of all bivaluations that satisfy $\mathcal{B}(\mathcal{L}, \bar{\theta})$.

Proof. Please see [9]. ☐

Example 2. *Recall the logic $\mathcal{L}$ introduced in Example 1. Note there are 3 unobtainable definite binary prints in the logic, more specifically $\langle T, T, F \rangle$, $\langle F, T, F \rangle$ and $\langle F, F, F \rangle$. The following is one of the three statements in the collection of U-statements associated to $\mathcal{L}$:*

$$(T : \varphi \quad \& \quad T : \bullet\varphi \quad \& \quad F : -\varphi) \quad \Longrightarrow \quad \bot \qquad \mathrm{U}\ \langle T, T, F \rangle$$

The fact that $R^{\theta_2 \bullet}_F = \{(4)\}$ reflects fact that if a bivaluation b_w satisfies $F : - \bullet (\varphi_1)$, then it must be the case that $w(\varphi_1) = 4$. Then the set $\Delta(4) = \{\langle F, T, T \rangle, \langle F, F, T \rangle\}$ expresses all the possible definite binary prints subformula φ_1 could take. The following equation expresses the B-statement associated with value F, connective $\bullet$ and separator $-$.

$$F : - \bullet \varphi \quad \Longrightarrow \quad (F : \varphi \quad \& \quad T : \bullet\varphi \quad \& \quad T : -\varphi) \quad \| \quad (F : \varphi \quad \& \quad F : \bullet\varphi \quad \& \quad T : -\varphi) \qquad \mathrm{B}^{\theta_2 \bullet}_F$$

Let $v = 4$. From the operator interpretation we know that $\widehat{\bullet}(4) = \{3, 4\}$. The general binary prints of these values are $\bar{\theta}(3) = \langle \{T\}, \{F\}, \{F\} \rangle$ and $\bar{\theta}(4) = \langle \{F\}, \{T, F\}, \{T\} \rangle$. Then the corresponding possible definite binary prints are $\Xi[\widehat{\bullet}(\bar{4})] = \{\langle T, F, F \rangle, \langle F, T, T \rangle, \langle F, F, T \rangle\}$. Therefore $\langle T, T, T \rangle$ is an $(\bullet, \bar{v})$-unobtainable binary print as in fact $\pi(\langle T, T, T \rangle) = 1$, and $1 \notin \widehat{\bullet}(\bar{v})$, and thus $\mathrm{N}^{\bullet \langle T, T, T \rangle}_{\langle F, T, T \rangle}$ is a statement of the logic:

$$(F : \varphi \quad \& \quad F : \bullet\, \varphi \quad \& \quad T : -\varphi \quad \& \quad T : \bullet\, \varphi \quad \& \quad T : \bullet\bullet\, \varphi \quad \& \quad T : -\bullet\, \varphi) \quad \Longrightarrow \quad \bot \qquad \mathrm{N}^{\bullet \langle T, T, T \rangle}_{\langle F, F, T \rangle}$$

3 Tableaux for non-deterministic many-valued logics

3.1 Dealing with partial information

The information at hand regarding the definite binary print of a formula may not always be maximal, that is, the information available may concern only some entries of a binary print. Therefore is of utmost importance to develop a way of recognizing the simplest unobtainable semantic scenarios and use that information in our favour.

Definition 14. Let a partial definite binary print be a sequence $\bar{Y} \in \langle F, T, \uparrow\rangle^{s+1}$. Strict partiality occurs when $\bar{Y} \notin \mathcal{V}_2{}^{s+1}$, as obviously otherwise $\bar{Y}$ is a total definite binary print. The set $dom(\bar{Y})$ is taken as $\{0 \leq r \leq s : Y_r \neq \uparrow\}$. Given partial definite binary prints $\bar{Y}$ and $\bar{Z}$, $\bar{Y}$ is said to extend $\bar{Z}$ if $dom(\bar{Z}) \subseteq dom(\bar{Y})$ and $Y_r = Z_r$ for every $r \in dom(\bar{Z})$. Strict extensions occur when $dom(\bar{Z}) \subset dom(\bar{Y})$, as any print, partial or total, is an extension of itself. A partial definite binary print $\bar{Y}$ is unobtainable if all of its possible $2^{s+1-|dom(\bar{Y})|}$ total extensions (which are total definite binary prints) are unobtainable. Otherwise it is called obtainable. An unobtainable partial binary print is said to be minimal if it is not an extension of another unobtainable partial binary print.

Although useful, the information given by the unobtainable total prints $\bar{X}$ can still be too general. In some cases information concerning a binary print, even if partial, inevitably leads to an absurd scenario. So we can take advantage of partial information available when it leads to unobtainable situations. The previous notation used can be extended to deal with partial definite binary prints. Given $\bar{Y} \in \{F, T, \uparrow\}^{s+1}$, consider the conjunction for arbitrary $\varphi \in \mathcal{S}$:

$$(\&_{r \in dom(\bar{Y})} Y_r : \theta_r(\varphi)). \qquad \text{(V } (\varphi; \bar{Y}))$$

In the same way as before, we can now introduce the following statement for minimal unobtainable print $\bar{Y}$ and arbitrary $\varphi \in \mathcal{S}$:

$$V(\varphi; \bar{Y}) \Longrightarrow \bot. \qquad (4)$$

Analogously to the case of U-statements, N-statements can also be improved to handle partial information. Instead of looking at the full information of the print of a formula and of its proper subformulas, one can look only at the print of the formula, and restrict be possible values of the proper subformulas. This statements will capture the behaviour of formulas $\theta_r(\odot(\varphi_1, ..., \varphi_k))$ in a similar way to B-statements. However the new statements we are about to define look at the total print of formula $\odot(\varphi_1, ..., \varphi_k)$, or the minimal generalized prints that entail a semantic impossibility not captured by B-statements, and thus do not suffer from the same shortcomings that led to the introduction of N-statements in the first place. First, we introduce the following set, for any operator $\odot \in \Sigma$ and partial binary print $\bar{Y}$:

$$\beta_{\bar{Y}}^{\odot} = \{\bar{v} \in \mathcal{V}^k : \bar{Y} \text{ is not minimal } (\odot, \bar{v})\text{-unobtainable}\}$$

. Given an operator $\odot \in \Sigma$ and an obtainable partial binary print $\bar{Y}$, for all values $\bar{v}$ in $\beta_{\bar{Y}}^{\odot}$ either:

1. An extension $\bar{X}$ of $\bar{Y}$ exists that $\bar{X} \in \Xi(\odot(v))$ or
2. There exists $r \in dom(\bar{Y})$ such that it is the case that $Y_r \neq \epsilon_r$ for every $\epsilon \in \Xi[\odot(v)]$.

Thus the right-hand side of this statements either captures a semantic possibility or an impossibility that is already expressed by B-statements. For each operator $\odot \in \Delta$ and obtainable general binary print Y, a new N-statement can be written as:

$$V(\varphi; \bar{Y}) \Rightarrow \|_{\bar{v} \in \beta_{\bar{Y}}^{\odot}} (\|_{\bar{\delta} \in \Delta(\bar{v})} V(\varphi_1, \ldots, \varphi_k; \bar{\delta})). \qquad (\text{N}_{Y}^{\odot})$$

Similarly to B-statements, the right-hand side of these statements can be reduced to their minimal equivalent boolean expression using minimization procedures. Please note the inclusion of elements satisfying case 2 above in the disjunction on the right-hand side of the equation is not necessary for the rule to be expressive enough, and may lead to greater branching, however in the cases where $\bar{Y}$ is not $(\odot, \bar{v})$-unobtainable for any $\bar{v} \in \mathcal{V}^k$, a tautological case of the rule is obtained, and the rule can be omitted altogether.

Definition 15. Let $\mathcal{B}_{\mathcal{T}}(\mathcal{L}, \bar{\theta})$ be the set of classic-like tableau statements associated to a separable non-deterministic finite-valued logic $\mathcal{L}$ with separating sequence $\bar{\theta} = \langle \theta_r \rangle_{r=0}^{s}$, which consists of all instances of:

- $U\bar{Y}$, for each minimal unobtainable partial definite binary print $\bar{Y}$, (U-Statements)
- $B_X^{\theta_r \odot}$, for each $X \in \{T, F\}$, $0 \leq r \leq s$ and $\odot \in \Sigma$, and (B-Statements)
- $N_{\bar{Y}}^{\odot}$, for each $\odot \in \Sigma_k$ and $\bar{Y} \in \{T, F, \uparrow\}^{ar(\odot)}$. ($N$-Statements)

Proposition 3. Sem_2 is the set of all bivaluations that satisfy $\mathcal{B}_{\mathcal{T}}(\mathcal{L}, \bar{\theta})$.

Proof. Please see [9]. □

Example 3. *Recall the logic $\mathcal{L}$ used throughout this text. None of the unobtainable binary prints $\langle T, T, F \rangle$, $\langle F, F, F \rangle$ and $\langle F, T, F \rangle$ is itself minimal. In fact they are extensions of (minimal) unobtainable partial binary prints $\langle \uparrow, T, F \rangle$ and $\langle F, \uparrow, F \rangle$, thus giving rise to revised U-statements:*

$$(T : \bullet\ \varphi \quad \& \quad F : -\varphi) \Longrightarrow \bot \qquad \text{U } \langle \uparrow, T, F \rangle$$

$$(F : \varphi \quad \& \quad F : -\varphi) \Longrightarrow \bot \qquad \text{U } \langle F, \uparrow, F \rangle$$

Recall that $N^{\bullet\langle T,T,T \rangle}_{\langle F,F,T \rangle}$ is a statement of the logic. Also, as seen before, $\langle T, T, T \rangle$ is a $(\bullet, 4)$-unobtainable binary print. In fact, the partial prints $\langle T, \uparrow, T, \rangle$ and $\langle T, T, \uparrow, \rangle$ are both minimal $(\bullet, 4)$-unobtainable binary prints. Therefore $4 \notin \beta^{\bullet}_{\langle T,\uparrow,T \rangle}$ and $4 \notin \beta^{\bullet}_{\langle T,T,\uparrow \rangle}$, and $\beta^{\bullet}_{\langle T,\uparrow,T} = \beta^{\bullet}_{\langle T,T,\uparrow \rangle}$={1,2,3}. This gives rise to the following two statements:

$$\begin{array}{c} (T : \bullet\ \varphi \quad \& \quad T : -\bullet\ \varphi) \quad \Longrightarrow \quad (T : \varphi \quad \& \quad T : \bullet\varphi \quad \& \quad T : -\varphi) \quad \| \\ (T : \varphi \quad \& \quad F : \bullet\varphi \quad \& \quad T : -\varphi) \quad \| \quad (T : \varphi \quad \& \quad F : \bullet\varphi \quad \& \quad F : -\varphi) \end{array} \qquad N^{\bullet}_{\langle T,\uparrow,T \rangle}$$

$$\begin{array}{c} (T : \bullet\ \varphi \quad \& \quad T : \bullet\bullet\ \varphi) \quad \Longrightarrow \quad (T : \varphi \quad \& \quad T : \bullet\varphi \quad \& \quad T : -\varphi) \quad \| \\ (T : \varphi \quad \& \quad F : \bullet\varphi \quad \& \quad T : -\varphi) \quad \| \quad (T : \varphi \quad \& \quad F : \bullet\varphi \quad \& \quad F : -\varphi) \end{array} \qquad N^{\bullet}_{\langle T,T,\uparrow \rangle}$$

3.2 Classic-like tableaux proof system

In order to develop the classic-like tableaux system, one final statement must be introduced. This statement expresses impossibility of a formula to be deemed non-exclusively true or false:

$$(F : \varphi \quad \& \quad T : \varphi) \Longrightarrow \bot \qquad \text{(ABS)}$$

It is evident that every bivaluation satisfies the statement. In fact, notice how due to their functional character no bivaluation satisfies the left-hand side of the statement. Each introduced statement S so far will univocally determine a branching rule in our classic-like tableaux system, which we will denote by $\mathcal{R}(S)$.

Definition 16. The classic-like tableau system $\mathcal{T}(\mathcal{L}, \bar{\theta})$ associated to logic $\mathcal{L}$ with the separating sequence $\bar{\theta}$ encloses the rules $\mathcal{R}(S)$ for every $S \in B_{\mathcal{T}}(\mathcal{L}, \bar{\theta})$, as well as the rule $\mathcal{R}(ABS)$.

Definition 17. A closure rule is any tableau rule whose conclusion is the absurd clause $\bot$.

Given a root set of classically-labelled formulas, the tableau is developed by applying time and again the set of rules to the formulas in the tableau. If the premises of some rule $\mathcal{R}$ are met by a branch of the tableau, then the application of the rule leads to the extension of that branch into as many branches as there are conclusions of $\mathcal{R}$. In the example presented every edge is going to be labelled with the statement that originated the rule application that gave birth to that branch (Figure 3.2).

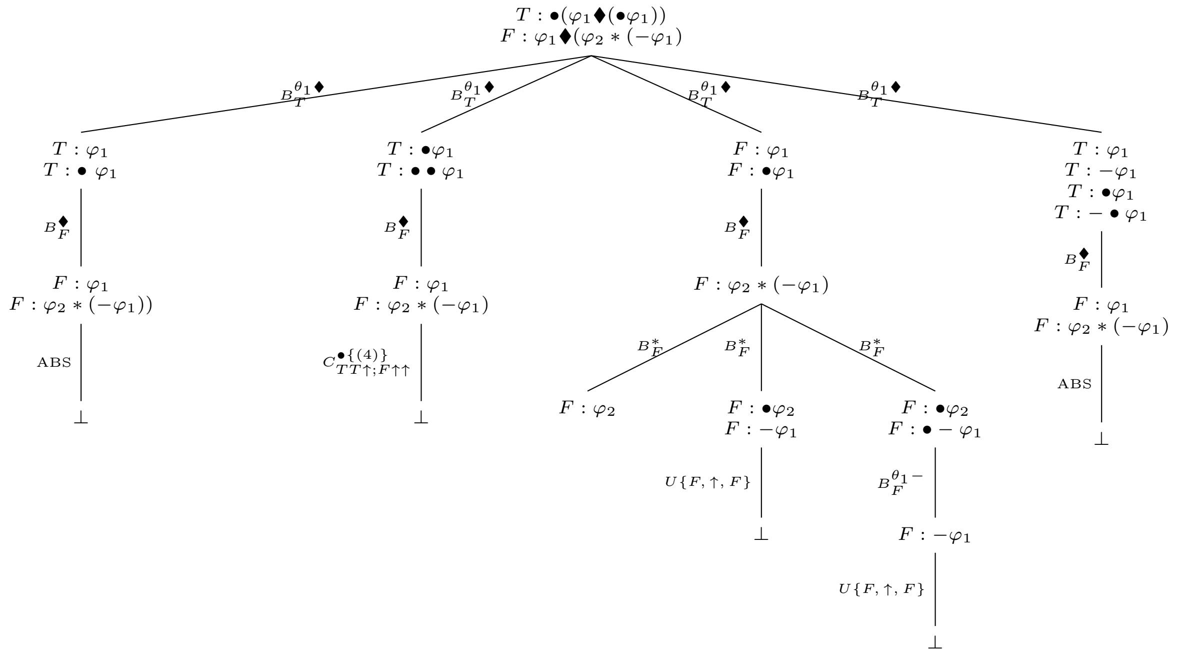

Figure 2: Open tableau for root set of formulas $T : \bullet(\varphi_1 \blacklozenge (\bullet\varphi_1))$ and $F : \varphi_1 \blacklozenge (\varphi_2 * (-\varphi_1))$ in $\mathcal{T}(\mathcal{L}^+, \bar{\theta} = \langle id, \bullet, - \rangle)$.

Definition 18. A branch is said to be closed if a closure rule has been applied, that is, if the branch contains $\bot$. If a branch is not closed it is called open. A tableau is closed if all its branches are closed, and open otherwise. A branch of a tableau is exhausted if it is either closed or if no rule can be applied to it. Naturally, a tableau is said to be exhausted if all its branches are exhausted.

The tableau system exposed so far does not guarantee termination by itself. In fact, it may happen that the application of a certain choice of rules leads to infinite branches. Additionally, signed formulas of the form $X : \theta_r(p)$, for $0 \le r \le s$ and $p \in \mathcal{A}$ already exhausted all information they could provide on the possible binary print of $w(p)$ for any valuation $w \in \mathsf{Sem}$. For that reason it is of utmost importance that we restrict the application of tableau rules in a way that ensures the termination of our method. To establish termination of the decision procedure a generalized concept of analiticity is imposed to the tableaux. For each node of a tableau, a formula $X : \theta_r(\varphi)$ can only be such that φ is a proper subformula of the original signed formulas.

A formula is analyzable if it is not of the form $X : \theta_r(p)$, where $X \in \{T, F\}$, $0 \le r \le s$ and p is an atom. Note that given a formula of the logic, it may match the left-hand side of more than one B-rule, because $\theta_{r_1}\odot_1$ and $\theta_{r_2}\odot_2$ may both be a fit. Without loss of generality assume the set of instances of statement $\theta_{r_1} \odot_1 (q_1, ..., q_{k_1})$ is a subset of the instances of statement $\theta_{r_2} \odot_2 (q_1, ..., q_{k_2})$. Then $\theta_{r_1}\odot_1$ is said to be more concrete than $\theta_{r_2}\odot_2$. A formula $\varphi \in S$ is called a proper $\theta_r\odot$-formula if $\theta_r\odot$ is the most concrete fit for φ.

Definition 19. Given $\varphi \in S$ the generalized notion of formula complexity is given by:

$$\mathsf{cplx}(\varphi) = \begin{cases} 0 & \text{if } \varphi \text{ is of the form } \theta_r(p) \text{ for some } p \in \mathcal{A} \\ 1 + \max\limits_{0 \le t \le s; 1 \le i \le k} \mathsf{cplx}(\theta_t(\varphi_i)) & \text{if } \varphi = \theta_r(\odot(\varphi_1, \ldots, \varphi_k)) \text{ is a proper } \theta_r \odot \text{-formula,} \\ & \quad \text{for } \odot \in \Sigma_k,\ k \neq 0,\ 0 \le r \le s, \text{ and } \varphi_1, \ldots \varphi_k \in \mathcal{S} \end{cases} \tag{5}$$

Definition 20. In the development of an analytic tableau in $\mathcal{T}(\mathcal{L}, \bar{\theta})$, the application of rules must conform to the following order:

1. closure rules;
2. most concrete applicable rule $\mathcal{R}(B_X^{\theta_r \odot})$ that has not yet been applied to the non exhausted analysable proper $\theta_r \odot$-formula of greater complexity in the tableau;
3. N-rules.

A branch of a tableau is called analytic if it is either closed or all possible rules have been applied according to the analytic proof-strategy introduced above. If all branches of a tableau are analytic, then the tableau is said to be analytic itself.

Proposition 4. Given a finite root set of classically-labelled formulas, a finite analytic tableau always exists.

Proof. Please see [9]. □

Proposition 5. If a valuation in Sem satisfies the initial root set of an (analytic) tableau, then it satisfies all the formulas in some open branch of the tableau.

Proof. Please see [9]. □

In the light of proposition 5, it is clear that when a root set of formulas originates a closed tableau, since any $b \in \mathsf{Sem}_2$ does not satisfy $\bot$, b cannot satisfy the root set itself, and therefore that set of formulas is unsatisfiable. However when the generated tableau is open, not only is that set satisfiable but the tableau is able to provide us with at least one valuation satisfying the classic-like root set.

Proposition 6. A valuation in Sem satisfying the root formulas can be extracted from every open branch of an (analytic) exhausted tableau.

Proof. Please see [9]. □

4 Web application

This Section reports on the development of a program to fully build a tableau proof-system for a sufficiently expressive logic. A compromise was made between efficiency and user friendly language with good code readability and comprehensive standard library, and the implementation was carried out in *Python*, a high-level object-oriented language, given its easy-to-use nature and intuitive syntax. The program comprises full tableau rules extraction for sufficiently expressive many-valued non-deterministic logics and tableau generation from a root-set of formulas, bound by a web application. The application was developed using open source web framework *Django*, with MVC(model-view-controller) design pattern, allowing for efficient code reuse and a faster, cleaner development. The application functionality is greatly divided in two windows. The first allows the user to specify a semantic, using a defined syntax and generate the corresponding set of rules. On the other hand the second page of the application allows the user to define the consequence relation $\varphi_1, ..., \varphi_k \vDash \phi_1, ..., \phi_k$ and test its validity by constructing a tableau from the application of rules calculated in the first page. This report does not go into detail on this matter. Please see [9] for more in-depth information regarding the implementation, features and use of the web application developed.

5 Conclusions

The main objective of this work, which was to provide a method for classic-like tableau generation aimed at non-determinisctic many-valued logics was achieved. Note there are several directions research on the present work can follow. At present statements characterizing the bivalent semantics are not extracted in their final form, as they are latter reduced/ simplified into streamlined versions. The method would benefit from optimized statement generation. Also the treatment of semantics of many-valued logics using only two truth-values comes at the price of significant complexity in tableau rules in terms of number of branches in rules. This ultimately may lead to undesirably large proofs. Thus one may contemplate the inclusion of the cut rule, as it is known proofs involving cuts are more efficient. In [10] and [7] the original work on which this work lies was used to develop a tableau proof system with analytic restrictions on the use of the cut rule for deterministic many-valued logics. Such system relies on linear rules for the connectives, with the cut rule being the sole branching rule. A similar direction could be taken to develop a more efficient tableaux system for non-deterministic many-valued logics. Lastly, a study on possible extensions of the present methodology could be explored in future work to allow application to other types of non-classic logics, as for example genuinely infinite-logics.

References

[1] A. Avron, B. Konikowska, and A. Zamansky. Cut-free sequent calculi for c-systems with generalized finite-valued semantics. *Journal of Logic and Computation*, 23:517–540, 2013.

[2] A. Avron and A. Zamansky. *Non-Deterministic Semantics for Logical Systems*, volume 16, pages 227–304. Springer, 2011.

[3] A. Avron and Anna Zammansky. Non-deterministic multi-valued logics - a tutorial. In *40th IEEE International Symposium on Multiple-Valued Logic*. IEEE, 20010.

[4] C. Caleiro, W. Carnielli, M. Coniglio, and João Marcos. Two's company: "the humbug of many logical values". In J.-Y. Béziau, editor, *Logica Universalis*, pages 169–189. Birkhäuser Verlag, 2005.

[5] C. Caleiro and J. Marcos. Classic-like analytic tableaux for finite-valued logics. In H. Ono, R. de Queiroz, and M. Kanazawa, editors, *WoLLIC 2009, Selected Papers*, volume 5514 of *Lecture Notes in Artificial Intelligence*, pages 268–280. Springer, 2009.

[6] C. Caleiro and J. Marcos. Two many values: An algorithmic outlook on suszko's thesis. In *Proceedings of the 40th Intl.Symp. on Many-Valued Logics (ISMVL 2010)*. IEEE Computer Society Press, 2010.

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

[8] Siegfried Gottwald. Many-valued logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2015 edition, 2015.

[9] Filipa Montez. Analytic classic-like tableaux for non-deterministic many-valued logics. Master's thesis, Instituto Superior Técnico, 2018.

[10] M. Volpe, J. Marcos, and C. Caleiro. Classic-like cut-based tableau systems for finite-valued logics. In L. Ong and R. de Queiroz, editors, *WoLLIC 2012, Selected Papers*, volume 7456 of *Lecture Notes in Computer Science*, pages 321–335. Springer, 2012.