Classic-Like Cut-Based Tableau Systems for Finite-Valued Logics

Marco Volpe[1], João Marcos[2], and Carlos Caleiro[3]

[1] Dipartimento di Informatica, Università di Verona, Italy
[2] LoLITA and DIMAp, UFRN, Brazil
[3] SQIG, Instituto de Telecomunicações and Depto. de Matemática, IST, Portugal

Abstract. A general procedure is presented for producing classic-like cut-based tableau systems for finite-valued logics. In such systems, cut is the only branching rule, and formulas are accompanied by signs acting as syntactic proxies for the two classical truth-values. The systems produced are guaranteed to be sound, complete and analytic, and they are also seen to polynomially simulate the truth-table method, thus extending the results in [7]. Łukasiewicz's 3-valued logic is used throughout as a simple illustrative example.

1 Introduction

In [4,5], in accordance with the so-called Suszko's Thesis, the authors have shown how to take advantage of the intrinsic bivalence that stems from the distinction between designated and undesignated truth-values in any sufficiently expressive finite-valued logic in order to provide the latter with a (non-truth-functional) bivalent semantics, and ultimately with a classic-like tableau proof system, using 2-signed formulas, associated to a simple decision procedure. However, due to the necessary encoding of the original semantics of the logic in terms of the two classical values, one ends up having to work with tableau rules having a significant number of branches that unavoidably lead to very large derivations.

It is widely known that proofs not involving cuts (or equivalently modus ponens) can be very inefficient. For classical propositional logic, for instance, cut-based proofs can be exponentially smaller than the shortest corresponding cut-free proofs (see [2]). Still, the unrestricted use of the cut rule poses a serious challenge for proof-search. First proposed by Mondadori, KE tableaux for classical logic, thoroughly studied in [6,9,7], are a cut-based tableau system that employs only analytic cuts and which is known to polynomially simulate the truth-table decision method, in the general case, bringing thus an exponential gain over conventional cut-free tableau systems in the worst cases.

Recent interest in KE tableaux (e.g. [10]) stimulated us to consider exploring a similar strategy, but now for producing classic-like cut-based tableau systems for finite-valued logics in general, capitalizing on [4,5], to which an analytic restriction of cut may be imposed, and which might in principle share the benefits of KE tableaux in terms of proof complexity. This paper reports on such an exploration.

L. Ong and R. de Queiroz (Eds.): WoLLIC 2012, LNCS 7456, pp. 321–335, 2012.

2 Background

Consider an alphabet with a denumerable set $\mathcal{A}$ of *atoms* and a finite set Σ of *primitive connectives*. The arity of a given connective $\odot \in \Sigma$ is to be denoted by $\mathsf{arg}\odot$. The set $\mathbb{S}$ of *formulas* is the carrier of the free Σ-algebra generated by $\mathcal{A}$. In dealing with finite-valued logics, $\mathcal{V}_n = \{v_0, v_1, \ldots, v_{n-1}\}$ will be used to denote sets of *truth-values*, given $n \in \mathbb{N}$, and these are supposed to be partitioned into a set $\mathcal{D} = \{v_i \mid 0 \leq i \leq m\}$ of *designated* values and a set $\mathcal{U} = \{v_i \mid m+1 \leq i \leq n-1\}$ of *undesignated* values. As a matter of stipulation, we will denote v_0 by F and v_{n-1} by T. In general, an (*n-valued*) *assignment* of truth-values to the atoms is any mapping $\rho : \mathcal{A} \to \mathcal{V}_n$, and an (*$n$-valued*) *valuation* is any extension $w : \mathbb{S} \to \mathcal{V}_n$ of such an assignment to the set of all formulas. An *n-valent semantics* for $\mathbb{S}$ based on $\mathcal{V}_n$, then, is simply a collection of n-valued valuations. In particular, we will call *bivalent* any (classic-like) semantics where $\mathcal{V}_2 = \{F, T\}$ and $\mathcal{D}_2 = \{T\}$; the corresponding valuations are called *bivaluations*. As usual, we call a valuation w a *model* of $\Delta \subseteq \mathbb{S}$ if $w(\Delta) \subseteq \mathcal{D}$. A canonical notion of *entailment* characterizing a *logic* $\mathcal{L}$ over $\mathbb{S}$ is associated to an n-valent semantics Sem by setting $\Gamma \models \alpha$ iff every model of Γ in Sem is a model of $\{\alpha\}$. A remarkable case of n-valent semantics corresponds to those we call *truth-functional*: such a semantics is given to the set of formulas $\mathbb{S}$ by defining an appropriate Σ-algebra $\mathbb{V}$ with carrier $\mathcal{V}_n$, by associating a k-ary interpretation operator $\widehat{\odot} : \mathcal{V}_n^k \to \mathcal{V}_n$ to each $\odot \in \Sigma$ with $\mathsf{arg}\odot = k$, and by collecting in Sem the set of all homomorphisms $\S : \mathbb{S} \to \mathbb{V}$. Any such homomorphism, as usual, may be understood as the unique extension of an assignment $\rho : \mathcal{A} \to \mathcal{V}_n$ into a valuation $\S_\rho : \mathbb{S} \to \mathbb{V}$ where $\S(\odot(\varphi_1, \ldots, \varphi_k)) = \widehat{\odot}(\S(\varphi_1), \ldots, \S(\varphi_k))$. Any logic characterized by truth-functional means, for a given $\mathcal{V}_n$, is called *n-valued*.

Let us now consider the total mapping $t : \mathcal{V}_n \to \mathcal{V}_2$ such that $t(v) = T$ iff $v \in \mathcal{D}$ and define, for any valuation $\S : \mathbb{S} \to \mathbb{V}$ of an n-valued semantics Sem, the bivaluation $b_\S = t \circ \S$. Though this bivalent semantics gives up the fundamental feature of truth-functionality, we have shown in previous papers (check [3] and the survey [5]) that it can still be very useful. As explained below, to accomplish the bivalent reduction constructively, in order to be able to distinguish any given value from any other value we just need to associate a unique 'binary print' to each truth-value of a given n-valued logic $\mathcal{L}$. Given $v_i, v_j \in \mathcal{V}$, we write $v_i \sharp v_j$ and say that v_i and v_j are *separated* in case $t(v_i) \neq t(v_j)$. Given two formulas φ_i and φ_j and a valuation $\S$ such that $v_i = \S(\varphi_i) \neq \S(\varphi_j) = v_j$ yet $b_\S(\varphi_i) = b_\S(\varphi_j)$, we say that a one-variable formula $\theta^{ij}(p)$ of $\mathcal{L}$ *separates* v_i and v_j if $\S(\theta^{ij}(\varphi_i)) \sharp \S(\theta^{ij}(\varphi_j))$ (or, equivalently, $b_\S(\theta^{ij}(\varphi_i)) \neq b_\S(\theta^{ij}(\varphi_j))$). In that case we will also say that the values v_i and v_j of $\mathcal{L}$ are *effectively distinguishable*, as they may be separated using the original linguistic resources of $\mathcal{L}$. Finally, we will say that the logic $\mathcal{L}$ is *effectively separable* in case its truth-values are pairwise effectively distinguishable, that is, for any pair of distinct values $\langle v_i, v_j \rangle \in \mathcal{D}^2 \cup \mathcal{U}^2$ a one-variable formula $\theta^{ij}(p)$ can be found in $\mathcal{L}$ that separates v_i and v_j. From this point on, for simplicity of exposition, we assume that all the necessary separators belong to the set Σ of primitive connectives of the logic — note that this is not really a restriction, as one can always conservatively extend an

n-valued logic $\mathcal{L}$ with a conveniently interpreted n-ary connective. Let Θ denote a finite sequence $[\theta_r(p)]_{r=0}^s = \langle\theta_0(p), \theta_1(p), \ldots, \theta_s(p)\rangle$ of one-variable formulas used as separators, where we assume $\theta_0(p) = p$. Obviously, $\theta_0(p)$ by itself suffices to separate any pair of values $\langle v_i, v_j\rangle \in (\mathcal{D} \times \mathcal{U}) \cup (\mathcal{U} \times \mathcal{D})$. We will call *binary print* of a value $v \in \mathcal{V}$ the sequence $\overline{v} = [b_\S(\theta_r(p))]_{r=0}^s$, where $\S(p) = v$. Notice that for every pair of distinct values $\langle v_i, v_j\rangle \in \mathcal{V}^2$ it is now obviously the case that $\overline{v_i} \neq \overline{v_j}$.

Example 1. Our running example will be Łukasiewicz's 3-valued logic, Ł$_3$. The logic may be described by choosing as primitive connectives $\Sigma = \{\neg, \diamond, \supset\}$, with $\mathsf{arg}\neg = \mathsf{arg}\diamond = 1$ *and* $\mathsf{arg}\supset = 2$*, and by considering the set of truth-values $\mathcal{V}_3 = \{v_0, v_1, v_2\}$, with v_2 as the sole designated value. The operators interpreting the connectives are described in Table 1.*

Table 1. Interpretation operators in Ł$_3$

x	$\widehat{\neg}x$	$\widehat{\diamond}x$
v_0	v_2	v_0
v_1	v_1	v_2
v_2	v_0	v_2

$x\widehat{\supset}y$	v_0	v_1	v_2
v_0	v_2	v_2	v_2
v_1	v_1	v_2	v_2
v_2	v_0	v_1	v_2

We need to look for a way of separating the two undesignated values v_0 and v_1, and accordingly we will have to set $\Theta = \langle p, \theta_1(p)\rangle$, for some convenient separator θ_1. There are two obvious separators already in the alphabet of Ł$_3$. We will here choose Łukasiewicz's 'possibility' operator $\diamond$ as θ_1. The same choice has in fact been made in [4], but there we have introduced $\diamond$ by abbreviation, noticing that $\widehat{\diamond}x \stackrel{def}{=} (\widehat{\neg}x)\widehat{\supset}x$. Clearly, such choice originates the binary prints $\langle F, F\rangle$, $\langle F, T\rangle$ and $\langle T, T\rangle$, respectively for v_0, v_1 and v_2. Note that the sequence $\langle T, F\rangle$ is unrealizable, as it does not correspond to any of the values in $\mathcal{V}_3$. Below, when $\diamond$ appears in the role of the separator θ_1 we will write it as θ, to help calling attention to the two different roles played by this connective. In [12] we have studied the effect of choosing Łukasiewicz's 'negation' operator $\neg$ as θ_1.

In earlier work, we have used this bivalent setting to produce classic-like tableau systems $\mathcal{T}(\mathcal{L}, \Theta)$ for any given n-valued logic $\mathcal{L}$ effectively separable by $\Theta = [\theta_r(p)]_{r=0}^s$. We refer the reader to [4,5] for the full details. However, it is worth mentioning here a few key ingredients of the procedure. Mirroring the classical truth-values $\{F, T\}$, we work with 2-signed formulas X:φ such that $\mathrm{X} \in \{\mathrm{F}, \mathrm{T}\}$ and $\varphi \in \mathbb{S}$. As a matter of convention, we shall say that an n-valued valuation $\S$ *satisfies* a labeled formula X:φ if $b_\S(\varphi) = X$. The notion of satisfaction extends naturally to sets of labeled formulas. Given a binary print $\overline{v} = [X_r]_{r=0}^s$, we use $\overline{v}^{\mathbb{S}}(\varphi)$ to denote the sequence of signed formulas $[\mathrm{X}_r{:}\varphi]_{r=0}^s$.

The cornerstone of $\mathcal{T}(\mathcal{L}, \Theta)$ is the recipe for obtaining elimination rules for the connectives. Using & to represent conjunction in the classical metalanguage, || to represent disjunction, $\Longrightarrow$ to represent implication, and $\circledast$ to represent an absurd, we produce a tableau rule for each schematic signed formula X:$\theta(\odot(\varphi_1, \ldots, \varphi_k))$ where $\mathrm{X} \in \{\mathrm{F}, \mathrm{T}\}$, $\theta \in \Theta$, and $\odot \in \Sigma$ with $\mathsf{arg}\odot = k$.

We further demand that if $\theta = \theta_0$, then $\odot \notin \Theta$, and we write more simply X: $\odot(\varphi_1, \ldots, \varphi_k)$ instead of X:$\theta_0(\odot(\varphi_1, \ldots, \varphi_k))$. The elimination rules are produced by collecting the tuples of binary prints that a homomorphic n-valuation § can assign to the formulas $\varphi_1, \ldots, \varphi_k$ in order to satisfy the signed formula. Letting $B^{\theta\odot}_{\mathrm{X}}([\varphi_i]^k_{i=1}) = \{\&[\overline{v_i}^{\S}(\varphi_i)]^k_{i=1} \mid t(\widehat{\theta}(\widehat{\odot}([v_i]^k_{i=1}))) = X\}$, the corresponding tableau rule is then given by

$$\mathrm{X}{:}\theta(\odot([\varphi_i]^k_{i=1})) \implies \| \, B^{\theta\odot}_{\mathrm{X}}([\varphi_i]^k_{i=1}).$$

In our metalanguage the above expression represents a tableau rule: the antecedent of each rule is the head, and the succedent describes the children nodes that may be created once the head matches a node of a previously given branch.

Example 2. In the case of Ł$_3$ with the single separator $\theta = \diamond$, the above described recipe would produce, for instance, a rule of the form

$$\mathrm{T}{:}\theta(\neg\varphi_1) \Longrightarrow (\mathrm{F}{:}\varphi_1 \;\&\; \mathrm{F}{:}\theta(\varphi_1)) \;\|\; (\mathrm{F}{:}\varphi_1 \;\&\; \mathrm{T}{:}\theta(\varphi_1))$$

simply because $\S(\diamond(\neg\varphi_1)) = v_2$ *if and only if* $\widehat{\diamond}(\widehat{\neg}(\S(\varphi_1))) = v_2$ *if and only if* $\S(\varphi_1) = v_0$ *or* $\S(\varphi_1) = v_1$. *Note that* $\langle F, F\rangle$ *and* $\langle F, T\rangle$ *are precisely the binary prints associated respectively to* v_0 *and* v_1.

Another example, now using the identity θ_0, *would yield*

$$\begin{aligned}\mathrm{F}{:}\varphi_1 \supset \varphi_2 \Longrightarrow \quad & (\mathrm{F}{:}\varphi_1 \;\&\; \mathrm{T}{:}\theta(\varphi_1) \;\&\; \mathrm{F}{:}\varphi_2 \;\&\; \mathrm{F}{:}\theta(\varphi_2)) \\ \| \; & (\mathrm{T}{:}\varphi_1 \;\&\; \mathrm{T}{:}\theta(\varphi_1) \;\&\; \mathrm{F}{:}\varphi_2 \;\&\; \mathrm{T}{:}\theta(\varphi_2)) \\ \| \; & (\mathrm{T}{:}\varphi_1 \;\&\; \mathrm{T}{:}\theta(\varphi_1) \;\&\; \mathrm{F}{:}\varphi_2 \;\&\; \mathrm{F}{:}\theta(\varphi_2))\end{aligned}$$

because $\S(\varphi_1 \supset \varphi_2) \neq v_2$ *if and only if* $\S(\varphi_1)\widehat{\supset}\S(\varphi_2) \neq v_2$ *if and only if* $\S(\varphi_1) = v_1$ *and* $\S(\varphi_2) = v_0$, *or* $\S(\varphi_1) = v_2$ *and* $\S(\varphi_2) = v_0$, *or* $\S(\varphi_1) = v_2$ *and* $\S(\varphi_2) = v_1$.

Such rules may be streamlined using classical equivalences in the metalanguage, and completeness of the tableau system is attained by the addition of suitable closure rules (see [4]).

As it might be expected, the tableau systems produced using the above recipe originate in general very redundant and highly branching derivations. The next sections will show how to use a similar approach to obtain more efficient systems, in which the only branching rule is an analytic version of the cut rule.

Before proceeding, we introduce some extra useful terminology and notation. As usual, each φ_i, for $1 \leq i \leq k$, is called an *immediate subformula* of $\odot(\varphi_1, \ldots, \varphi_k)$. The set of *proper subformulas* of a given $\odot(\varphi_1, \ldots, \varphi_k)$ contains the immediate subformulas of this formula and the immediate subformulas of any formula therein contained. We here dub *Θ-immediate subformula* of $\odot(\varphi_1, \ldots, \varphi_k)$ any formula of the form $\theta(\varphi_i)$, for $1 \leq i \leq k$ and $\theta \in \Theta$. The set of *proper Θ-subformulas* of a given formula has the obvious definition. A Θ-formula is called *atomic* if it has no Θ-immediate subformulas. We also define the *size* of a formula (signed or not) to be the cardinality of its set of subformulas (forgetting the sign, in the case of a signed formula). For convenience, we will assume $F^C = T$ and $T^C = F$ as the *conjugates* of the two classical truth-values, and extend the notation accordingly to the syntactic labels T and F.

In the next section we will illustrate the ideas behind our novel rule-extraction algorithm by discussing what happens in the running example of $Ł_3$. After that we will present and study our general method in full detail.

3 A Cut-Based Tableau System for $Ł_3$

The idea here is to find a suitable way of defining a tableau system for $Ł_3$ whose only branching rule is a cut rule, in a way that generalizes the KE tableaux of [6,9], proposed for classical logic. Recall that we consider $Ł_3$ separators $\Theta = \langle p, \theta(p)\rangle$, where $\theta = \diamond$. Our tableau system will consist of three classes of rules: the cut rule, elimination rules, and closure rules.

The cut rule is the only branching rule, i.e., the only rule with more than one branch in the succedent, and has the following typical form:

$$(Ł_3.\text{Cut}) \qquad \Longrightarrow \text{F}{:}\varphi \;\;||\;\; \text{T}{:}\varphi$$

In Section 4 we will show that it is possible to restrict its use only to analytic applications.

We will now take full advantage of the classic-like semantics of $Ł_3$ introduced by its corresponding bivalent semantics, obtained following the procedure detailed in [3], and extract from it suitable elimination and closure rules for our novel cut-based system.

As explained and illustrated in Section 2, we will need suitable elimination rules for signed formulas of the forms $\text{X}{:}\neg\varphi_1$, $\text{X}{:}\varphi_1 \supset \varphi_2$, $\text{X}{:}\theta(\neg\varphi_1)$, $\text{X}{:}\theta(\varphi_1 \supset \varphi_2)$ and $\text{X}{:}\theta(\diamond(\varphi_1))$, where $\theta = \diamond$ and $\text{X} \in \{\text{F}, \text{T}\}$. Recall that, given a formula φ, we can express its 3-valued truth-table as a bivalent one, where the value of φ depends only on the values of its Θ-subformulas. Given that the procedure is systematic, let us focus at a fragment of it, and consider the bivalent version of

Table 2. The bivalent version of $\supset$

combination	φ_1	$\theta(\varphi_1)$	φ_2	$\theta(\varphi_2)$	$\varphi_1 \supset \varphi_2$
0	F	F	F	F	T
1	F	F	F	T	T
2	F	F	T	F	–
3	F	F	T	T	T
4	F	T	F	F	F
5	F	T	F	T	T
6	F	T	T	F	–
7	F	T	T	T	T
8	T	F	F	F	–
9	T	F	F	T	–
10	T	F	T	F	–
11	T	F	T	T	–
12	T	T	F	F	F
13	T	T	F	T	F
14	T	T	T	F	–
15	T	T	T	T	T

the truth-table corresponding to the formula $\varphi_1 \supset \varphi_2$. In Table 2 we include all the combinations for the signs of φ_1, $\theta(\varphi_1)$, φ_2, $\theta(\varphi_2)$. A dash (–) in the last column indicates that the corresponding combination contains a sequence $\langle T, F\rangle$ for some $\langle\varphi, \theta(\varphi)\rangle$ that corresponds to no binary print $\overline{v}$, for $v \in \mathcal{V}_3$.

From Table 2 we can mechanically extract a set of elimination rules for Ł$_3$'s 'implication' connective $\supset$. Indeed, consider the partial bivaluation b^j described at combination j of the table, in such a way that we shall say that $\mathrm{X}_j{:}\psi$ is satisfied if ψ is at the head of some column and the j-th combination below it contains value X_j. In our cut-based tableau system there will be a rule corresponding to each collection R of signed formulas satisfied by some partial bivaluation b_j with the requirement that such collection must contain $\mathrm{X}_j{:}\varphi_1 \supset \varphi_2$. For instance, some possible such collections are $\{\mathrm{F}{:}\varphi_1 \supset \varphi_2\}$, $\{\mathrm{F}{:}\varphi_1 \supset \varphi_2, \mathrm{T}{:}\varphi_1\}$ and $\{\mathrm{T}{:}\varphi_1 \supset \varphi_2, \mathrm{F}{:}\theta(\varphi_1), \mathrm{T}{:}\theta(\varphi_2)\}$. Each such collection R, read as a conjunction, will form the antecedent of a tableau rule. Let $\mathsf{Mod}(R)$ be the set of all partial bivaluations corresponding to combinations that satisfy R. The succedent of the corresponding rule will contain the (possibly empty) collection, read as a conjunction, of all signed formulas that are simultaneously satisfied by all the bivaluations in $\mathsf{Mod}(R)$. As an example, let $\{\mathrm{F}{:}\varphi_1 \supset \varphi_2\}$ be the antecedent of a given rule. Then we can restrict our attention to the combinations 4, 12 and 13 from Table 2. We may easily notice that $\{\mathrm{T}{:}\theta(\varphi_1), \mathrm{F}{:}\varphi_2\}$ is an invariant in these combinations. The corresponding tableau rule will in this case read:

$$(Ł_3.\!\supset 1^*) \qquad \mathrm{F}{:}\varphi_1 \supset \varphi_2 \Longrightarrow \mathrm{T}{:}\theta(\varphi_1) \;\&\; \mathrm{F}{:}\varphi_2.$$

Note that we omit the (empty) rules originating from partial bivaluations for which in the derived restricted table we have no invariants (other than the signed formulas fixed for the antecedent). For example, we do not have any rule with $\{\mathrm{T}{:}\varphi_1 \supset \varphi_2\}$ as antecedent, since $\mathrm{T}{:}\varphi_1 \supset \varphi_2$ itself is the only invariant in the corresponding restricted table (it suffices to contrast combinations 0 and 15). A general and formal account of this rule-extraction procedure will be given in Section 4. Table 3 contains the full set of rules obtained, in particular, for the connective $\supset$.

It is clear that the procedure described above for the mechanical extraction of elimination rules may generate a lot of redundancies. As a trivial example, one may notice that the rule $(Ł_3.\!\supset 2^*)$ of Table 3 is redundant in the presence of $(Ł_3.\!\supset 1^*)$ since they have the same succedent and the collection of antecedents of one of them is included in the other. One may notice that the rule $(Ł_3.\!\supset 4^*)$ is also redundant in the presence of $(Ł_3.\!\supset 1^*)$, given that the latter has a more informative succedent than the former, even if it contains less hypotheses in the antecedent. After the elimination of all such redundant rules, and repeating the procedure for all connectives, with and without the separator θ, we obtain the elimination rules in Table 4.

Finally, with respect to the closure rules, we follow [4] to the letter. Besides the traditional closure rule for 2-signed tableaux, which says that a branch is closed once it contains two signed formulas of the form $\mathrm{F}{:}\varphi$ and $\mathrm{T}{:}\varphi$, additional closure rules will be needed in order to exclude unrealizable binary prints — in

Table 3. Rules automatically derived from the truth-table for $\supset$

$(\text{Ł}_3.\!\supset 1^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 2^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 3^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 4^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 5^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 6^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 7^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 8^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 9^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 10^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 11^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 12^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 13^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 14^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 15^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 16^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 17^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{F}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 18^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 19^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 20^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 21^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 22^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 23^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 24^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 25^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 26^*)$	$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 27^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 28^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 29^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 30^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 31^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 32^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 33^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 34^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 35^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 36^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 37^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 38^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 39^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 40^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 41^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 42^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_1$
$(\text{Ł}_3.\!\supset 43^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 44^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 45^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\varphi_2 \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 46^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 47^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{F}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 48^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{F}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 49^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 50^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{F}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 51^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{F}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 52^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 53^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 54^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 55^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_1)$
$(\text{Ł}_3.\!\supset 56^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1)$	$\Longrightarrow$	$\mathrm{T}{:}\varphi_2 \ \& \ \mathrm{T}{:}\theta(\varphi_2)$
$(\text{Ł}_3.\!\supset 57^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\theta(\varphi_2)$	$\Longrightarrow$	$\mathrm{T}{:}\varphi_2$
$(\text{Ł}_3.\!\supset 58^*)$	$\mathrm{T}{:}\varphi_1 \supset \varphi_2 \ \& \ \mathrm{T}{:}\varphi_1 \ \& \ \mathrm{T}{:}\theta(\varphi_1) \ \& \ \mathrm{T}{:}\varphi_2$	$\Longrightarrow$	$\mathrm{T}{:}\theta(\varphi_2)$

this case of Ł$_3$ and Θ, we are talking about $\langle T, F\rangle$. Hence, an additional closure rule will say that branches containing both a signed formula of the form T:φ and a signed formula of the form F:$\theta(\varphi)$ may be closed. One might represent the above mentioned such closure rules by writing:

$$\begin{array}{lll}(Ł_3.\mathrm{C0}) & \mathrm{F}{:}\varphi \ \& \ \mathrm{T}{:}\varphi & \Longrightarrow \circledast \\ (Ł_3.\mathrm{C1}) & \mathrm{T}{:}\varphi \ \& \ \mathrm{F}{:}\theta(\varphi) & \Longrightarrow \circledast\end{array}$$

Table 4. Streamlined elimination rules of the tableau system for Ł$_3$

(Ł$_3$.$\neg$1)	F:$\neg\varphi_1$	$\Longrightarrow$ T:$\theta(\varphi_1)$
(Ł$_3$.$\neg$2)	T:$\neg\varphi_1$	$\Longrightarrow$ F:φ_1 & F:$\theta(\varphi_1)$
(Ł$_3$.$\supset$ 1)	F:$\varphi_1 \supset \varphi_2$	$\Longrightarrow$ T:$\theta(\varphi_1)$ & F:φ_2
(Ł$_3$.$\supset$ 2)	F:$\varphi_1 \supset \varphi_2$ & T:$\theta(\varphi_2)$	$\Longrightarrow$ T:φ_1 & T:$\theta(\varphi_1)$ & F:φ_2
(Ł$_3$.$\supset$ 3)	F:$\varphi_1 \supset \varphi_2$ & F:φ_1	$\Longrightarrow$ T:$\theta(\varphi_1)$ & F:φ_2 & F:$\theta(\varphi_2)$
(Ł$_3$.$\supset$ 4)	T:$\varphi_1 \supset \varphi_2$ & F:$\theta(\varphi_2)$	$\Longrightarrow$ F:φ_1 & F:$\theta(\varphi_1)$ & F:φ_2
(Ł$_3$.$\supset$ 5)	T:$\varphi_1 \supset \varphi_2$ & F:φ_2	$\Longrightarrow$ F:φ_1
(Ł$_3$.$\supset$ 6)	T:$\varphi_1 \supset \varphi_2$ & T:φ_2	$\Longrightarrow$ T:$\theta(\varphi_2)$
(Ł$_3$.$\supset$ 7)	T:$\varphi_1 \supset \varphi_2$ & F:$\theta(\varphi_1)$	$\Longrightarrow$ F:φ_1
(Ł$_3$.$\supset$ 8)	T:$\varphi_1 \supset \varphi_2$ & F:$\theta(\varphi_1)$ & T:φ_2	$\Longrightarrow$ F:φ_1 & T:$\theta(\varphi_2)$
(Ł$_3$.$\supset$ 9)	T:$\varphi_1 \supset \varphi_2$ & T:$\theta(\varphi_1)$	$\Longrightarrow$ T:$\theta(\varphi_2)$
(Ł$_3$.$\supset$ 10)	T:$\varphi_1 \supset \varphi_2$ & T:$\theta(\varphi_1)$ & F:φ_2	$\Longrightarrow$ F:φ_1 & T:$\theta(\varphi_2)$
(Ł$_3$.$\supset$ 11)	T:$\varphi_1 \supset \varphi_2$ & T:φ_1	$\Longrightarrow$ T:$\theta(\varphi_1)$ & T:φ_2 & T:$\theta(\varphi_2)$
(Ł$_3$.$\theta\neg$1)	F:$\theta(\neg\varphi_1)$	$\Longrightarrow$ T:φ_1 & T:$\theta(\varphi_1)$
(Ł$_3$.$\theta\neg$2)	T:$\theta(\neg\varphi_1)$	$\Longrightarrow$ F:φ_1
(Ł$_3$.$\theta \supset$ 1)	F:$\theta(\varphi_1 \supset \varphi_2)$	$\Longrightarrow$ T:φ_1 & T:$\theta(\varphi_1)$ & F:φ_2 & F:$\theta(\varphi_2)$
(Ł$_3$.$\theta \supset$ 2)	T:$\theta(\varphi_1 \supset \varphi_2)$ & F:$\theta(\varphi_2)$	$\Longrightarrow$ F:φ_1 & F:φ_2
(Ł$_3$.$\theta \supset$ 3)	T:$\theta(\varphi_1 \supset \varphi_2)$ & T:φ_2	$\Longrightarrow$ T:$\theta(\varphi_2)$
(Ł$_3$.$\theta \supset$ 4)	T:$\theta(\varphi_1 \supset \varphi_2)$ & F:$\theta(\varphi_1)$	$\Longrightarrow$ F:φ_1
(Ł$_3$.$\theta \supset$ 5)	T:$\theta(\varphi_1 \supset \varphi_2)$ & F:$\theta(\varphi_1)$ & T:φ_2	$\Longrightarrow$ F:φ_1 & T:$\theta(\varphi_2)$
(Ł$_3$.$\theta \supset$ 6)	T:$\theta(\varphi_1 \supset \varphi_2)$ & T:φ_1	$\Longrightarrow$ T:$\theta(\varphi_1)$ & T:$\theta(\varphi_2)$
(Ł$_3$.$\theta(\diamond 1)$)	F:$\theta(\diamond(\varphi_1))$	$\Longrightarrow$ F:φ_1 & F:$\theta(\varphi_1)$
(Ł$_3$.$\theta(\diamond 2)$)	T:$\theta(\diamond(\varphi_1))$	$\Longrightarrow$ T:$\theta(\varphi_1)$

Figure 1 shows an example of a tableau for Ł$_3$ using the set of rules obtained as described above. In this example we get (2.1) and (2.2) by applying rule (Ł$_3$.$\supset$1) to the formula (1). The same rule applies to (2.2) to originate (3.1) and (3.2). An application of (Ł$_3$.$\supset$3) to (1) and (3.2) gives (4.1). Then we apply (Ł$_3$.$\theta \supset$1) to (4.1) and get (5.1) and (5.2). We close the tableau by applying (Ł$_3$.C0) to (2.1) and (5.2). Note that the derivation tree is linear as no use of (Ł$_3$.Cut) was necessary.

$$\begin{array}{c}
(1)\ \mathrm{F}{:}(p_0 \supset (p_1 \supset p_0)) \\
| \\
(2.1)\ \mathrm{T}{:}\theta(p_0) \\
(2.2)\ \mathrm{F}{:}(p_1 \supset p_0) \\
| \\
(3.1)\ \mathrm{T}{:}\theta(p_1) \\
(3.2)\ \mathrm{F}{:}p_0 \\
| \\
(4.1)\ \mathrm{F}{:}\theta(p_1 \supset p_0) \\
| \\
(5.1)\ \mathrm{T}{:}p_1 \\
(5.2)\ \mathrm{F}{:}\theta(p_0) \\
| \\
\circledast
\end{array}$$

Fig. 1. A refutation of $p_0 \supset (p_1 \supset p_0)$ in the cut-based tableau system for Ł$_3$

4 The Tableau System

4.1 Rules

Let $\mathcal{L}$ be an effectively separable n-valued logic with a set of formulas $\mathbb{S}$ generated over the set of connectives Σ by the set of atoms $\mathcal{A}$, and having $\mathcal{D} \subseteq \mathcal{V}_n$ as its set of designated values. We assume also that its binary prints are produced by a convenient sequence of separators $\Theta = [\theta_r(p)]_{r=0}^{s}$, where $\theta_0(p) = p$. In the following, we will exhibit the rules of our novel cut-based tableau system for $\mathcal{L}$.

As explained before, the only branching rule of our system is:

$$(\mathcal{L}.\text{Cut}) \qquad\qquad \Longrightarrow \text{F:}\varphi \;\;||\;\; \text{T:}\varphi$$

Below in this section, we will show that it is possible to restrict the use of such cut rule only to analytic applications, that is, applications to tableau branches of which φ is a Θ-subformula.

Let now $\mathsf{BP} = \{F, T\}^{s+1}$ be the set of all $(s+1)$-long binary prints and let a *partial binary print* be any sequence $\overline{c}_R = [c_r]_{r\in R}$ such that $R \subseteq \{0, 1, \ldots, s\}$ and each $c_r \in \{F, T\}$ (this definition includes, of course, all binary prints in BP, as strict partiality occurs precisely when R is a proper subset of $\{0, 1, \ldots, s\}$). We say that a partial binary print $\overline{d}_U$ *extends* $\overline{c}_R$ if $R \subseteq U$ and $d_r = c_r$ for every $r \in R$.

We say that a sequence $[\overline{v_i}]_{i=1}^k$ of binary prints *satisfies* a signed formula $\text{X:}\theta(\odot([\varphi_i]_{i=1}^k))$ if $t(\widehat{\theta}(\widehat{\odot}([v_i]_{i=1}^k))) = X$. Further, we say that a signed formula is *satisfiable by a sequence* $[\overline{c_i}_{R_i}]_{i=1}^k$ *of partial binary prints* if it is satisfied by some sequence of binary prints that extends $[\overline{c_i}_{R_i}]_{i=1}^k$ componentwise.

Let $R_i, U_i \subseteq \{0, 1, \ldots, s\}$ be such that $R_i \cap U_i = \varnothing$, for each $1 \le i \le k$, let $[\overline{c_i}_{R_i}]_{i=1}^k$ and $[\overline{d_i}_{U_i}]_{i=1}^k$ be two disjoint sequences of partial binary prints, and let δ be the signed formula $\text{X:}\theta(\odot([\varphi_i]_{i=1}^k))$. We say that $[\overline{c_i}_{R_i}]_{i=1}^k$ *entails* $[\overline{d_i}_{U_i}]_{i=1}^k$ *with respect to* δ when, for every sequence $[\overline{v_i}]_{i=1}^k$ of binary prints satisfying δ, if $[\overline{v_i}]_{i=1}^k$ extends $[\overline{c_i}_{R_i}]_{i=1}^k$ then $[\overline{v_i}]_{i=1}^k$ extends $[\overline{d_i}_{U_i}]_{i=1}^k$.

We now produce elimination rules for each signed formula $\delta = \text{X:}\theta(\odot([\varphi_i]_{i=1}^k))$ such that if $\theta = \theta_0$, then $\odot \notin \Theta$. We consider, for each sequence of partial binary prints $[\overline{c_i}_{R_i}]_{i=1}^k$ that satisfies δ, the following rule:

$$(\mathcal{L}.R_{\text{X}}^{\theta\odot}[\overline{c_i}_{R_i}]_{i=1}^k) \qquad \text{X:}\theta(\odot([\varphi_i]_{i=1}^k))\;\&\;(\&[\overline{c_i}^{\mathbb{S}}_{R_i}(\varphi_i)]_{i=1}^k) \Longrightarrow \&[\overline{d_i}^{\mathbb{S}}_{U_i}(\varphi_i)]_{i=1}^k$$

where $[\overline{d_i}_{U_i}]_{i=1}^k$ is the largest sequence of partial binary prints entailed by $[\overline{c_i}_{R_i}]_{i=1}^k$ with respect to δ. That is to say that $\overline{d_i}_{U_i}$ extends any other sequence of partial binary prints entailed by $[\overline{c_i}_{R_i}]_{i=1}^k$ with respect to δ. Note that such a largest partial binary print is well-defined. Indeed, given the fact that δ is satisfiable, any two entailed sequences of partial binary prints $[\overline{e_i}_{V_i}]_{i=1}^k$ and $[\overline{f_i}_{W_i}]_{i=1}^k$ are compatible, i.e., for each i, if $j \in V_i \cap W_i$ then $e_{ij} = f_{ij}$, and can thus be joined into $[\overline{g_i}_{V_i\cup W_i}]_{i=1}^k$ such that, for each i, $g_{ij} = e_{ij}$ if $j \in V_i$ and $g_{ij} = f_{ij}$ if $j \in W_i$. Clearly, $[\overline{g_i}_{V_i\cup W_i}]_{i=1}^k$ extends both sequences and is also entailed by $[\overline{c_i}_{R_i}]_{i=1}^k$ with respect to δ.

The set of elimination rules listed above might contain a lot of redundancies. We can see an elimination rule as a pair of sets $\langle \Pi_1, \Pi_2 \rangle$ where Π_1 contains the signed formulas in the antecedent and Π_2 the signed formulas in the succedent of the rule. In this case, we say that a rule (Δ_1, Δ_2) is *redundant* in a system $\mathcal{T}$ if there is a different rule (Γ_1, Γ_2) in $\mathcal{T}$ such that: (i) $\Gamma_1 \subseteq \Delta_1$; and (ii) $\Delta_2 \subseteq \Gamma_2$.

Finally, closure rules look precisely as in the system of [5]. We briefly explain the procedure below, for the sake of self-containment.

We consider first the usual classic-like closure rule:

$$(\mathcal{L}.\mathrm{C0}) \qquad\qquad \mathrm{F}{:}\varphi \;\&\; \mathrm{T}{:}\varphi \Longrightarrow \circledast$$

In addition, we have to consider the unrealizable binary prints. Let $\mathsf{CS} = \mathsf{BP} \setminus \{\overline{v} \mid v \in \mathcal{V}_n\}$ be the set of all the bivalent sequences that are *not* produced as binary prints of truth-values of $\mathcal{L}$. Intuitively, any *closuring sequence* $\overline{c} \in \mathsf{CS}$ brings about information that is unobtainable, allowing one thus to close a tableau branch that contains it. Information, even if partial, leading unambiguously to a sequence in CS should always give rise to a closed tableau. Indeed, closuring information is carried by any partial binary print $\overline{c}_R$ such that all of its $2^{\#(\Theta)-\#(R)}$ possible total extensions are in CS. Hence, it would be reasonable to add a different closure rule for each such partial closuring information. However, it suffices to take into account just the minimal closuring situations, that is, closuring partial sequences $\overline{c}_R$ that cannot be obtained as extensions of any other closuring partial sequence. In general, where $\overline{c}_R = [c_r]_{r \in R}$ is some partial binary print, we write $\overline{c}^{\mathbb{S}}_R(\varphi) = [s(c_r){:}\theta_r(\varphi)]_{r \in R}$ for the linguistic 2-signed version of such sequence, where $s(c_r) = \mathrm{T}$ if $c_r = T$ and $s(c_r) = \mathrm{F}$ if $c_r = F$. Accordingly, for each minimal closuring partial binary print $\overline{c}_R$, we consider an additional closure rule:

$$(\mathcal{L}.\mathrm{C}k) \qquad\qquad \&\big(\overline{c}^{\mathbb{S}}_R(\varphi)\big) \Longrightarrow \circledast$$

Finally, we get further closure rules as particular cases in the production of elimination rules. Namely, we need to consider when the formula $\mathrm{X}{:}\theta(\odot([\varphi_i]_{i=1}^k))$ is not satisfiable. For any such a case, we consider the additional closure rule:

$$(\mathcal{L}.\mathrm{C}^{\theta\odot}_{\mathrm{X}}) \qquad\qquad \mathrm{X}{:}\theta(\odot([\varphi_i]_{i=1}^k)) \Longrightarrow \circledast$$

We can now define our full cut-based tableau system.

Definition 1. *The tableau system* $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ *for the logic* $\mathcal{L}$ *with respect to* Θ *is composed of rule* $(\mathcal{L}.\mathrm{Cut})$*, non-redundant elimination rules* $(\mathcal{L}.R^{\theta\odot}_{\mathrm{X}}[\overline{c_i}_{R_i}]_{i=1}^j)$*, and closure rules* $(\mathcal{L}.\mathrm{C0})$*,* $(\mathcal{L}.\mathrm{C}k)$*,* $(\mathcal{L}.\mathrm{C}^{\theta\odot}_{\mathrm{X}})$ *defined as above.*

Tableaux are built as usual, by applying the above rules, given an initial sequence of 2-signed formulas, and a branch is said to be *closed* if its closure is obtained by the application of any of the $(\mathrm{C}k)$ rules, including $(\mathrm{C0})$, or of any $C^{\theta\odot}_{\mathrm{X}}$ rule. Branches that are not closed are said to be *open*. A tableau is said to be *closed* in case all of its branches are closed.

4.2 Properties

We will now check the soundness and completeness of our cut-based tableau systems $\mathcal{T}^{\mathsf{cut}}(\mathcal{L},\Theta)$.

As usual, we say that the system is *sound* if the root of any closed tableau is unsatisfiable. Conversely, we say that the system is *complete* if every unsatisfiable finite set of signed formulas is the root of some closed tableau.

Theorem 1. *The tableau system $\mathcal{T}^{\mathsf{cut}}(\mathcal{L},\Theta)$ is sound and complete.*

Proof. For soundness, it is sufficient to show that if a homomorphic n-valuation $\S : \mathbb{S} \to \mathcal{V}_n$ satisfies the head of a rule then it must satisfy one of the branches of its succedent. This is clearly the case for the cut rule. The property also holds for the closure rules, as shown in [4,5]. We are thus left with proving the claim for the linear elimination rules $(\mathcal{L}.R^{\theta\odot}_{\mathrm{X}}[\overline{c_i}_{R_i}]^k_{i=1})$, which holds basically by construction. Indeed, if $\S$ satisfies $\mathrm{X}{:}\theta(\odot([\varphi_i]^j_{i=1}))$ and $[\overline{c_i}^{\mathbb{S}}_{R_i}(\varphi_i)]^k_{i=1}$ then $\S$ must also satisfy $[\overline{d_i}^{\mathbb{S}}_{U_i}(\varphi_i)]^k_{i=1}$ because $[\overline{c_i}^{\mathbb{S}}_{R_i}(\varphi_i)]^k_{i=1}$ entails $[\overline{d_i}^{\mathbb{S}}_{U_i}(\varphi_i)]^k_{i=1}$ with respect to $\mathrm{X}{:}\theta(\odot([\varphi_i]^j_{i=1}))$.

We prove completeness of $\mathcal{T}^{\mathsf{cut}}(\mathcal{L},\Theta)$ by exploiting the completeness of the tableau system $\mathcal{T}(\mathcal{L},\Theta)$ defined in [4,5]. Clearly, it is enough to show that all the rules of $\mathcal{T}(\mathcal{L},\Theta)$ are derivable in $\mathcal{T}^{\mathsf{cut}}(\mathcal{L},\Theta)$. Closure rules are common to both systems. Thus, we just need to show that it is possible to simulate in $\mathcal{T}^{\mathsf{cut}}(\mathcal{L},\Theta)$ the branching elimination rules of $\mathcal{T}(\mathcal{L},\Theta)$, extracted as explained in Section 2. Let us pick one arbitrary such rule

$$\mathrm{X}{:}\theta(\odot([\varphi_i]^k_{i=1})) \implies \|\ B^{\theta\odot}_{\mathrm{X}}([\varphi_i]^k_{i=1})$$

where we recall that $B^{\theta\odot}_{\mathrm{X}}([\varphi_i]^k_{i=1}) = \{\&[\overline{v_i}^{\mathbb{S}}(\varphi_i)]^k_{i=1} \mid t(\widehat{\theta}(\widehat{\odot}([v_i]^k_{i=1}))) = X\}$.

Given the root $\mathrm{X}{:}\theta(\odot([\varphi_i]^k_{i=1}))$, we start by using $(\mathcal{L}.\mathrm{Cut})$ to cut on all the immediate Θ-subformulas of $\odot([\varphi_i]^k_{i=1})$. This will produce $2^{k\cdot\#(\Theta)}$ branches corresponding to all possible combinations of classical values for $\theta_j(\varphi_i)$ with $j = 0, 1, \ldots, s$ and $i = 1, \ldots, k$. The branches that correspond to combinations that satisfy the head of the rule coincide precisely with the elements of $B^{\theta\odot}_{\mathrm{X}}([\varphi_i]^k_{i=1})$. Thus we are left with showing that the remaining branches can all be closed. Some of these branches may close simply by means of an application of some $(\mathcal{L}.\mathrm{C}k)$ rule because they correspond to combinations that include some unrealizable binary print (as the dashed lines in Table 2). Hence, we only need to analyze what happens with the branches corresponding to valid combinations that assign the value X^C to $\theta(\odot([\varphi_i]^k_{i=1}))$. Consider the sequence of elements in one such branch and take its largest prefix that turns $\mathrm{X}{:}\theta(\odot([\varphi_i]^k_{i=1}))$ satisfiable. It is, of course, a proper prefix. Assume also that $\mathrm{Y}{:}\theta_j(\varphi_i)$ is the next element in the sequence. Clearly, the prefix corresponds to some sequence $[\overline{c_i}_{R_i}]^k_{i=1}$ of partial binary prints whose associated rule $(\mathcal{L}.R^{\theta\odot}_{\mathrm{X}}[\overline{c_i}_{R_i}]^k_{i=1})$ will produce $\mathrm{Y}^C{:}\theta_j(\varphi_i)$ (or a simpler rule if this one is redundant). Finally, we may close the branch using the rule $(\mathcal{L}.\mathrm{C0})$. $\square$

The strategy used in the completeness proof above is simple but often builds unnecessarily complex tableaux. Below, when we study the proof complexity of our cut-based systems, we will show that such tableaux can be significantly simplified. In any case, most importantly, the proof of Theorem 1 also shows the completeness of the analytic version of our cut-based systems, i.e., a restriction that allows applications of cut only to Θ-subformulas of the formulas occurring in the root of the tree.

Corollary 1. *The analytic restriction of $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ is complete.*

In the light of the analyticity result in Corollary 1, the cut-based tableau system $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ can be used as a decision procedure for the logic $\mathcal{L}$. Since finite-valued logics are already known to be decidable by the 'brute force' truth-table method, it will be interesting to know more about the computational complexity of the decision procedure associated to $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$. As in the case of the KE system for classical logic (see [6]), it is expectable that our cut-based tableaux for finite-valued logics fare significantly better than conventional tableaux in terms of proof complexity, and in general not worse than the truth-table method. We adapt from [8] the definition of some typical complexity measures to be used below.

Definition 2. *The* size *of a tableau π, denoted by $|\pi|$ is the total number of formulas occurring in π. The* λ-complexity *of a tableau π, denoted by $\lambda(\pi)$, is the number of nodes in π.The* ρ-complexity *of a tableau π, denoted by $\rho(\pi)$, is the maximum number of formulas in a node of π.*

As an example, for the tableau π in Figure 1, we have $|\pi| = 9$, $\lambda(\pi) = 6$ and $\rho(\pi) = 2$. Clearly, the following relation holds in general: $|\pi| \leq \lambda(\pi) \cdot \rho(\pi)$. Note that in the case of a tableau π produced within $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$, the ρ-complexity of π is bounded by $\rho(\pi) \leq k(s+1)$, where $s+1$ is the cardinality of Θ and k is the maximum arity of any connective from the alphabet of $\mathcal{L}$.

The following theorem shows that the cut-based tableau systems given by Definition 1 can polynomially simulate (p-simulate) the truth-table method.

Theorem 2. *Given a valid signed formula $\mathrm{X}{:}\varphi$ of $\mathcal{L}$ with size a and containing v distinct atoms, there is a refutation π of $\mathrm{X}^C{:}\varphi$ in $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ of complexity $\lambda(\pi) = O(a \cdot \#(\Theta) \cdot 2^{v \cdot \#(\Theta)})$.*

Proof. First we apply ($\mathcal{L}$.Cut) to all the atomic Θ-subformulas of φ. This will generate a tree with $2^{v \cdot \#(\Theta)}$ branches. Then, for each such branch, we proceed by applying ($\mathcal{L}$.Cut) to a Θ-subformula φ_i of φ such that all of its immediate Θ-subformulas are already in the branch. By construction, such a φ_i exists. We note that at least one of the two branches thereby generated gives rise to a contradiction and may be closed by applying at most one elimination rule and one closure rule. Indeed, by the definition of the system, either the system contains an elimination rule for φ_i whose application gives rise to a contradiction on one of the Θ-subformulas of φ_i or, as a trivial case, φ_i is of the form $\theta(\odot(\dots))$ and we can apply a closure rule ($\mathcal{L}.\mathrm{C}^{\theta\odot}_{\mathrm{X}}$), that is, either $\mathrm{F}{:}\varphi_i \Longrightarrow \divideontimes$ or $\mathrm{T}{:}\varphi_i \Longrightarrow \divideontimes$. If one of the branches does not close, we can reiterate on it the same procedure, by applying ($\mathcal{L}$.Cut) to a further Θ-subformula of φ such that all its immediate Θ-subformulas are in the branch.

We conclude by noticing that all the initial $2^{v \cdot \#(\Theta)}$ branches may be closed by following the above described procedure, i.e., by applying ($\mathcal{L}$.Cut) to at most the Θ-subformulas of φ, and so linearly in $a \cdot \#(\Theta)$. $\square$

We can further show that $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ is not worse than $\mathcal{T}(\mathcal{L}, \Theta)$. Intuitively, we must be able to reproduce efficiently in $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ any tableau constructed within $\mathcal{T}(\mathcal{L}, \Theta)$, and in particular more efficiently than we managed to do in the proof of Theorem 1. To illustrate how we proceed, we show in particular how it is possible to efficiently simulate in the cut-based tableau system for $Ł_3$ (Section 3) the branching rule for F:$p \supset q$ (Example 2). While the tree on the left of Figure 2 portrays an application of the rule obtained in Section 2, the one on the right represents its efficient simulation by means of rules of the cut-based system. In particular, we use ($Ł_3.\supset 1$) to derive (2.1) and (2.2); then we cut on p and obtain (3.1) and (3.2); finally, we obtain (4) by using ($Ł_3.\supset 3$) on (1) and (3.1) and we obtain (5.1) and (5.2) by cutting on $\theta(q)$.

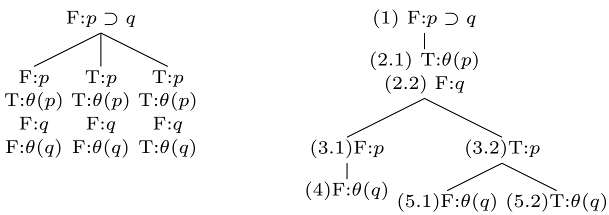

Fig. 2. Finding efficient simulations of branching elimination rules for $Ł_3$

The proof of the following theorem uses a similar strategy.

Theorem 3. *For every proof π in the system $\mathcal{T}(\mathcal{L}, \Theta)$, there exists a proof π^{cut} with the same root in the system $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ such that $|\pi^{\mathsf{cut}}| \leq |\pi|$.*

Proof. Building upon the proof of Theorem 1, it is enough to show that each branching elimination rule of $\mathcal{T}(\mathcal{L}, \Theta)$ can be efficiently derived in the cut-based system. Let us consider an arbitrary such rule

$$\mathrm{X}{:}\theta(\odot([\varphi_i]_{i=1}^k)) \implies || \; B_{\mathrm{X}}^{\theta\odot}([\varphi_i]_{i=1}^k)$$

where $B_{\mathrm{X}}^{\theta\odot}([\varphi_i]_{i=1}^k) = \{\&[\overline{v_i}^{\mathbb{S}}(\varphi_i)]_{i=1}^k \mid t(\widehat{\theta}(\widehat{\odot}([v_i]_{i=1}^k))) = X\}$, as in Section 2.

Starting with root X:$\theta(\odot([\varphi_i]_{i=1}^k))$, in $\mathcal{T}^{\mathsf{cut}}(\mathcal{L}, \Theta)$ we can follow a procedure consisting in applying linear elimination rules for X:$\theta(\odot([\varphi_i]_{i=1}^k))$ whenever possible, or ($\mathcal{L}$.Cut) on some missing Θ-subformula if none of the elimination rules can be applied. It is easy to see that, by construction, the amount of information in the simulating tree is not bigger than the one produced by the given rule, i.e., each formula in such a simulating tree also occurs in at least one branch of the rule. $\square$

The existence of extremely bad cases, in general, for $\mathcal{T}(\mathcal{L}, \Theta)$ is very likely, although exploring that path lies beyond the scope of this paper. Together with the above results, one would then certainly expect to be able to show, as in the case of classical logic, that the cut-based systems allow in general for a significantly better performance.

5 Conclusions

Other paths could have been explored for defining appropriate cut-based versions of the tableau systems in [4,5]. Yet, we believe that the path explored here achieves a good trade-off between efficiency of proof construction and usability of the system. On what concerns the first aspect, as it is common in this area, we measured efficiency in terms of size of the tableaux produced, by having in mind, as a minimum requirement, that p-simulation of truth-tables must hold. Clearly, the use of a larger number of rules would help in this sense; in particular, we could add a closure rule for each unsatisfiable situation arising from the analysis of truth-tables, as illustrated in Section 3 and formalized in Section 4. This would in principle reduce —but asymptotically not in any significant way— the size of the closed tableaux built as in the proof of Theorem 2, since each unsatisfiable branch could be closed immediately. A further option would consist in allowing only elimination rules such that all the immediate subformulas are involved in the rule, either in the antecedent or in the succedent. As an example, the rule ($Ł_3.\supset 1$) would not be allowed in the system of Section 3. The systems resulting from such approach allow for the p-simulation of the truth-table method (the procedure described in the proof of Theorem 2 can still be applied) and have the advantage of facilitating proof search, in the sense that for each formula in a tableau one needs to apply at most one elimination rule. A drawback of such systems is that they tend to require more uses of cut, e.g., the formula in the example of Figure 1 (see Section 3) would not have a linear closed tableau.

On what concerns readability and compactness of the system, we mainly tried to minimize the number of rules and the number of formulas per rule. With such goal in mind, further simplifications could be proposed. As an example, one can notice that the rule ($Ł_3.\supset 2$) might be rewritten as

$$\mathrm{F}{:}\varphi_1 \supset \varphi_2 \ \&\ \mathrm{T}{:}\theta(\varphi_2) \quad \Longrightarrow \quad \mathrm{T}{:}\varphi_1$$

since the other formulas in the succedent may be obtained by an application of ($Ł_3.\supset 1$). By generalizing such simplifications, one would obtain a more compact system for which, however, the result of Theorem 2 would not hold. Finally, we note that the proof of Theorem 2 suggests a very simple decision procedure, which is enough for p-simulating truth-tables. However, in general there might be better heuristics for guiding the construction of a tableau. For example, the canonical procedure given in [8] for the KE system for classical logic coincides, in essence, with the procedure we adopted in the proof of Theorem 3.

As we have seen, the syntactic encoding of the truth-tabular semantics presupposed by our classic-like approach to cut-based tableaux generates in principle a multiplication of the number of rules. Moreover, in the resulting tableau systems, rules contain a number of expressions in the antecedent which need to be simultaneously matched to the nodes of a given branch in order to be applied. Even though proof-complexity theorists do not in general take into account the costs implicit in the use of a deductive system with a large number of rules and with rules which require a lot of pattern-matching effort, and we have here done our study in accordance with that tradition, one might also think it wiser to measure such costs in calculating the efficiency of a given proof system.

Though our methods cannot be expected to apply to infinite-valued logics in general, it is predictable that they extend smoothly at least to those infinite-valued logics with a finite-valued non-deterministic semantics [1]. The possible connection between our approach and resolution-based sets-as-signs methods [11] is another interesting topic for future research.

Acknowledgments. The third author thanks the support of FCT and FEDER via the projects PEst-OE/EEI/LA0008/2011 and ComFormCrypt PTDC/EIA-CCO/113033/2009 of IT, and UTAustin/MAT/0057/2008 of IST, as well as of the PQDR and GeTFun initiatives of SGIQ. The authors are indebted to five anonymous referees for their careful reading of an earlier version of this paper.

References

1. Avron, A., Lev, I.: Non-deterministic multiple-valued structures. Journ. of Logic and Computation 15, 241–261 (2005)
2. Boolos, G.: Don't eliminate cut. Journ. of Phil. Logic 13, 373–378 (1984)
3. Caleiro, C., Carnielli, W., Coniglio, M.E., Marcos, J.: Two's company: The humbug of many logical values. In: Béziau, J.-Y. (ed.) Log. Universalis, pp. 169–189. Birkhäuser Verlag, Basel (2005)
4. Caleiro, C., Marcos, J.: Classic-Like Analytic Tableaux for Finite-Valued Logics. In: Ono, H., Kanazawa, M., de Queiroz, R. (eds.) WoLLIC 2009. LNCS, vol. 5514, pp. 268–280. Springer, Heidelberg (2009)
5. Caleiro, C., Marcos, J.: Many-valuedness meets bivalence: Using logical values in an effective way. J. of Multiple-Valued Log. and Soft Comp. 18 (2012)
6. D'Agostino, M.: Investigations into the complexity of some propositional calculi. PRG Techn. Monogr. 88. Oxford Univ., Computing Lab, Oxford (1990)
7. D'Agostino, M.: Are tableaux an improvement on truth-tables? Cut-free proofs and bivalence. Journ. of Log., Lang., and Inform. 1, 235–252 (1992)
8. D'Agostino, M.: Tableau methods for classical propositional logic. In: Handbook of Tableau Methods, pp. 45–123. Kluwer Academic Publishers (1999)
9. D'Agostino, M., Mondadori, M.: The taming of the cut: Classical refutations with analytic cut. Journ. of Log. and Comp. 4(3), 285–319 (1994)
10. Finger, M., Gabbay, D.: Cut and pay. Journ. of Logic, Language and Information 15, 195–218 (2006)
11. Hähnle, R.: Automated Deduction in Multiple-Valued Logics. International Series of Monographs on Computer Science, vol. 10. Oxford University Press (1994)
12. Marcos, J., Mendonça, D.: Towards fully automated axiom extraction for finite-valued logics. In: Carnielli, W., et al. (eds.) The Many Sides of Logic, pp. 425–440. College Publications, London (2009)