**L**ogic and **L**ogical **P**hilosophy
Volume 28 (2019), 223–257
DOI: 10.12775/LLP.2018.009

**Yaroslav Petrukhin**
**Vasilyi Shangin**

# AUTOMATED PROOF-SEARCHING FOR STRONG KLEENE LOGIC AND ITS BINARY EXTENSIONS VIA CORRESPONDENCE ANALYSIS*

**Abstract.** Using the method of correspondence analysis, Tamminga obtains sound and complete natural deduction systems for all the unary and binary truth-functional extensions of Kleene's strong three-valued logic $\mathbf{K_3}$. In this paper, we extend Tamminga's result by presenting an original finite, sound and complete proof-searching technique for all the truth-functional binary extensions of $\mathbf{K_3}$.

**Keywords**: proof search; correspondence analysis; three-valued logic; strong Kleene logic; natural deduction; proof theory

## 1. Introduction and motivation

Taking their inspiration from modal correspondence theory [44, 59, 60], Kooi and Tamminga [26] presented correspondence analysis for the unary and binary extensions of the three-valued logic of paradox **LP** [2, 42, 41]. Correspondence analysis allows one to immediately find inference rules for the unary and binary operators added to **LP** from their truth tables' entries. Let $\star$ be an arbitrary unary or binary operator added to **LP** and $f_\star$ be its truth table. Then an inference scheme $\Pi \vdash \phi$ is said to characterize an $f_\star$'s entry $\mathscr{E}$ iff the following condition holds: $\Pi \models \phi$

* The first author is supported by Polish National Science Centre, grant number DEC-2017/25/B/HS1/01268. The second author is supported by Russian Foundation for Humanities, grant 16-03-00749 "Logical-epistemic problems of knowledge representation".

*Received* September 15, 2017. *Revised* February 8, 2018. *Published online* July 6, 2018

iff $\mathscr{E}$ is an $f_\star$'s entry. So, all $f_\star$'s entries are characterized by inference schemes. Moreover, these inference schemes are inference rules, in fact. If one adds them to a natural deduction system for **LP**, one obtains a sound and complete natural deduction system for **LP** extended by $\star$. Automated proof-searching for natural deduction systems for **LP** and its extensions is presented in [36]. In this paper, we will deal with the correspondence analysis for the extensions of Kleene's strong three-valued logic $\mathbf{K_3}$ [25, 24] which was presented by Tamminga [55]. Based upon Tamminga's paper, we develop an original sound and complete proof-searching technique in the spirit of [10, 36] for the natural deduction systems for the binary extensions of $\mathbf{K_3}$.[1]

Natural deduction successfully overcame skepticism concerning its appropriateness to automated theorem-proving [13]. See, for example, [52, 20]. In the paper, we aren't going to outline the topic. However, we stress the following aspects concerning our motivation. First, the paper deals with automated proof-searching for Fitch-style natural deduction for strong Kleene logic, and, to the best of our knowledge, this is original. Strong Kleene logic is itself worth studying for modelling partial recursive predicates and reasoning in a situation with a lack of information (the third value is interpreted as 'intermediate' or 'unknown'). Second, there are arguments that favour a Jaśkowski-Fitch style of natural deduction to Gentzen style of natural deduction [17]. Both reasons make proof-searching for natural deduction in this logic an important task. Moreover, we go further by proposing an original proof-searching procedure not only for this logic but for its generalizations which include some prominent logics mentioned in the literature (see sections 1.3 and 1.4 further). And the correctness argument (finiteness, soundness, and completeness) for the proof-searching procedure in all of these logics is given in one go. Last, but not least, the importance of the logics in question being equipped with a correct proof-searching procedure lies within the paradigm by Dov Gabbay, which strongly suggests, roughly speaking, that if the same logic has two different proof-searching procedures, then it's not the same logic [14].

The structure of this paper is as follows: in the rest of this section, we briefly introduce a semantics for $\mathbf{K_3}$ (subsection 1.1), the notion

[1] Although Tamminga [55] presented correspondence analysis for the unary and binary extensions of $\mathbf{K_3}$, we will deal with the binary ones only. Note that correspondence analysis was also adapted for four-valued logics [33].

of correspondence analysis (subsection 1.2), and discuss some of $\mathbf{K_3}$'s notable binary extensions (subsections 1.3 and 1.4); in section 2, we consider natural deduction systems for $\mathbf{K_3}$ and its binary extensions; in section 3, we present our proof-searching technique; in section 4, we show how it works, using some examples; in section 5, we prove its soundness, completeness and termination; in section 6, we discuss related work; and in section 7, we summarize our results.

### 1.1. Semantics of strong Kleene logic

Strong Kleene logic $\mathbf{K_3}$ is built over a propositional language $\mathscr{L}$ which contains a set $\mathscr{P} = \{p, q, r, s, p_1, \ldots\}$ of propositional variables, left and right parentheses, negation ($\neg$), disjunction ($\vee$), and conjunction ($\wedge$). A set $\mathscr{F}$ of all $\mathscr{L}$-formulas is defined in a standard way. A set $\mathscr{V}$ of truth values contains the following elements only: 1 (true), i (intermediate/unknown), 0 (false). 1 (true) is the designated value. A function $f_\star$ is said to be a truth table $f$ for an operator $\star$.[2] A valuation $v$ is said to be a function from $\mathscr{P}$ to $\mathscr{V}$. A valuation $v$ on $\mathscr{P}$ is extended to a valuation on $\mathscr{F}$ as follows:

| $\phi$ | $f_\neg$ |
|---|---|
| 1 | 0 |
| i | i |
| 0 | 1 |

| $f_\vee$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| i | 1 | i | i |
| 0 | 1 | i | 0 |

| $f_\wedge$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | i | 0 |
| i | i | i | 0 |
| 0 | 0 | 0 | 0 |

DEFINITION 1. Let $\Pi \subseteq \mathscr{L}$ and $\phi \in \mathscr{L}$. Then $\Pi \models \phi$ is defined as follows: if $v(\psi) = 1$, for all $\psi \in \Pi$, then $v(\phi) = 1$, for each valuation $v$.

### 1.2. The notion of correspondence analysis

$\mathbf{K_3^\circ}$ ($\mathbf{K_3}$'s extension by binary operators $\circ_1, \ldots, \circ_n$) is built over $\mathscr{L}^\circ$ ($\mathscr{L}$'s extension by $\circ_1, \ldots, \circ_n$). Throughout the paper, $\circ$ denotes an arbitrary element of $\{\circ_1, \ldots, \circ_n\}$ and $f_\circ(x, y) = z$ denotes an $f_\circ$'s entry such that for all $x, y, z \in \{0, \mathrm{i}, 1\}$ and all $\phi, \psi \in \mathscr{L}$ if $v(\phi) = x$ and $v(\psi) = y$, then $v(\phi \circ \psi) = z$, for each valuation $v$. Though the notion of an entry of a truth table seems to be obvious, let us clarify that an entry is neither a row nor a column in a truth table. In the truth tables above, an entry of a truth table for $\star$ is any cell containing a valuation of $\star$. For instance, the truth table for negation contains exactly three entries.

[2] $f_\star$ takes an operator $\star$ as an input and outputs with a truth table for $\star$.

Note that since $\mathbf{K_3}$ is not functionally complete, it is not the case that $f_{\circ_1}, \ldots, f_{\circ_n}$ are definable via $f_\neg$, $f_\wedge$, $f_\vee$.

We will use the following adaptation of Kooi and Tamminga's definitions 2.1 and 1 from [26] and [55], respectively.

DEFINITION 2 (Single Entry Correspondence [26, 55]). Let $\Pi \cup \{\phi\} \subseteq \mathscr{L}^\circ$. Let $x, y, z \in \{0, \mathrm{i}, 1\}$. Then the truth table entry $f_\circ(x, y) = z$ is characterized by an inference scheme $\Pi \vdash \phi$, if

$$f_\circ(x, y) = z \text{ if and only if } \Pi \models \phi.$$

Tamminga [55, Theorem 1] found inference schemes which characterize an $f_\circ$'s all possible entries (see Theorem 1 in Section 2 below).

### 1.3. Implicational extensions of $\mathbf{K_3}$

Recall that an $\circ$ is an arbitrary binary truth-functional operator. In particular cases, an $\circ$ may be an implication. So, in this section, we consider some notable examples of $\mathbf{K_3}$'s implicational extensions.

| $f_{\to_1}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | i | 0 |
| i | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

| $f_{\to_2}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | i | 0 |
| i | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

| $f_{\to_3}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | i | 0 |
| i | 1 | 1 | i |
| 0 | 1 | 1 | 1 |

| $f_{\to_4}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| i | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

| $f_{\to_5}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| i | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

| $f_{\to_6}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| i | 1 | 1 | i |
| 0 | 1 | 1 | 1 |

| $f_{\to_7}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| i | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |

| $f_{\to_8}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| i | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

Let us start with Heyting's implication (we denote it by $\to_1$). It originally appeared in Heyting's logic $\mathbf{G_3}$ that was studied by Heyting [19], Gödel [15], and Jaśkowski [21]. However, $\mathbf{G_3}$'s negation is not the same as $\mathbf{K_3}$'s one. Although $\mathbf{G_3}$ is not an implicational extension of $\mathbf{K_3}$, Batens [5] presented a logic which is $\mathbf{K_3}$ extended by Heyting's implication. Natural deduction for $\mathbf{G_3}$ itself is presented in [34].

Słupecki's implication $\to_2$ was presented in [53] and [31] as an attempt to restore the deduction theorem to Łukasiewicz's $\mathbf{Ł_3}$ [28]. $\mathbf{K_3}$ that was extended by Słupecki's implication appeared untitled in Avron's paper [3]. Following Popov [38], we call this logic **PComp**.

The abovementioned $\mathbf{Ł_3}$ [28] is $\mathbf{K_3}$ extended by Łukasiewicz's implication $\to_3$.

Although Rescher's logic [43] is not an implicational extension of $\mathbf{K_3}$, Rescher's $\to_4$ can be added to $\mathbf{K_3}$ as an implicational connective.

Bochvar's $\to_5$ is an implication of his logic $\mathbf{B_3}$ [8]. Note that $\to_5$ is also an implication of Sette and Carnielli's weakly intuitionistic logic $\mathbf{I_1}$ [46] (which is a dual of Sette's $\mathbf{P_1}$ [45]) and Popov's (and Marcos') $\mathbf{I_2}$ [39, 29] (which, in turn, is a dual of Marcos' $\mathbf{P_2}$ [29]). Although none of the abovementioned logics containing $\to_5$ is an implicational extension of $\mathbf{K_3}$, one may consider $\mathbf{K_3}$ that was extended by $\to_5$.

All these implications are natural in the sense of Tomova [56]. Moreover, as follows from [56], in the case of the only designated value 1, there are only 6 natural implications $\to_i$, $1 \leqslant i \leqslant 6$. A natural deduction characterisation (via correspondence analysis) of natural logics can be found in [37].

However, in [58, 57] Tomova presents an extended class of natural implications. In the case of the only designated value 1, this class contains implications $\to_i$, $1 \leqslant i \leqslant 8$. Note that $\to_7$ is an implication of Karpenko and Tomova's literal paralogic $\mathbf{TK^2}$ [22] ($\mathbf{TK^1}$'s implication is $\to_4$). Moreover, $\to_8$ is Sette's implication [45].

### 1.4. Peirce's arrow and Sheffer's stroke as extensions of $\mathbf{K_3}$

However, it is not the case that an $\circ$ may be implication only. It may be either Peirce's arrow or Sheffer's stroke. In [50], Shestakov introduced Peirce's arrow $\downarrow_1$ for $\mathbf{K_3}$. Moreover, in [49] and [51], Shestakov presented Peirce's arrows $\downarrow_2$ and $\downarrow_3$ for $\mathbf{B_3}$'s internal and external connectives, respectively. In [30], McKinsey introduced Sheffer's stroke for $\mathbf{Ł_3}$ ($\Downarrow$).

| $f_{\Downarrow}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 0 | i | 1 |
| i | i | 1 | 1 |
| 0 | 1 | 1 | 1 |

| $f_{\downarrow_1}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| i | 0 | i | i |
| 0 | 0 | i | 1 |

| $f_{\downarrow_2}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 0 | i | 0 |
| i | i | i | i |
| 0 | 0 | i | 1 |

| $f_{\downarrow_3}$ | 1 | i | 0 |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| i | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |

## 2. Natural deduction systems

Natural deduction systems for $\mathbf{K_3}$ were presented independently by Priest [41] and by Tamminga [55].[3] Tamminga's system $\mathfrak{ND}_{\mathbf{K_3}}$ contains the following inference rules.

*Elimination rules*:

$$(\wedge\mathrm{E}_1)\ \frac{\phi \wedge \psi}{\phi} \qquad (\wedge\mathrm{E}_2)\ \frac{\phi \wedge \psi}{\psi} \qquad (\vee\mathrm{E})\ \frac{\phi \vee \psi \quad \begin{matrix}[\phi]\\ \chi\end{matrix} \quad \begin{matrix}[\psi]\\ \chi\end{matrix}}{\chi}$$

$$(\neg\neg\mathrm{E})\ \frac{\neg\neg\phi}{\phi} \qquad (\neg\vee\mathrm{E})\ \frac{\neg(\phi \vee \psi)}{\neg\phi \wedge \neg\psi} \qquad (\neg\wedge\mathrm{E})\ \frac{\neg(\phi \wedge \psi)}{\neg\phi \vee \neg\psi}$$

*Introduction rules*:

$$(\wedge\mathrm{I})\ \frac{\phi \quad \psi}{\phi \wedge \psi} \qquad (\vee\mathrm{I}_1)\ \frac{\phi}{\phi \vee \psi} \qquad (\vee\mathrm{I}_2)\ \frac{\psi}{\phi \vee \psi} \qquad (\neg\neg\mathrm{I})\ \frac{\phi}{\neg\neg\phi}$$

$$(\neg\vee\mathrm{I})\ \frac{\neg\phi \wedge \neg\psi}{\neg(\phi \vee \psi)} \qquad (\neg\wedge\mathrm{I})\ \frac{\neg\phi \vee \neg\psi}{\neg(\phi \wedge \psi)} \qquad (\mathrm{EFQ})\ \frac{\phi \quad \neg\phi}{\psi}$$

Although in [41, 55] a natural deduction derivation is defined in a tree-format, we will define it in a linear-format (sometimes referred to as Jaśkowski-Fitch style), following Copi, Cohen, and McMahon's textbook [11, p. 366].[4]

DEFINITION 3. A derivation in $\mathfrak{ND}_{\mathbf{K_3}}$ of a formula $\omega$ from a set of assumptions $\Pi$ is a finite nonempty sequence of formulae with the following conditions:

1. Each formula is an assumption or follows from the previous formulae via an $\mathfrak{ND}_{\mathbf{K_3}}$-rule;
2. By applying ($\vee$E) each formula starting from the assumption $\phi$ until a formula $\chi$, inclusively, as well as each formula starting from the assumption $\psi$ until a formula $\chi$, inclusively, is discarded from the derivation.

A proof in $\mathfrak{ND}_{\mathbf{K_3}}$ is a derivation from the empty set of assumptions. Note that although both a derivation and a proof cannot be empty, later we will show the way our proof-searching procedure deals with the

[3] We follow Tamminga's formulation; however, the only difference between Tamminga's and Priest's calculi is with regard to the rule (EFQ). In Priest's version it is as follows: $\frac{\phi \wedge \neg\phi}{\psi}$.

[4] See also [48], where a precise definition of a derivation is discussed.

situation when no derivation is found and, moreover, the sequence of formulae after proof-searching is empty (see Sections 3 and 4).

As follows from [41] and [55], $\mathfrak{ND}_{\mathbf{K_3}}$ is sound and complete.

To obtain a natural deduction system $\mathfrak{ND}_{\mathbf{K_3^\circ}}$ for $\mathbf{K_3^\circ}$ we need the following theorem.

THEOREM 1 (Tamminga [55]). *For all* $\phi, \psi, \chi \in \mathscr{L}^\circ$ *we have*:

$$f_\circ(0,0) = \begin{cases} 0 & \textit{iff} \quad \neg\phi \wedge \neg\psi \models \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \neg\phi \wedge \neg\psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \chi \\ 1 & \textit{iff} \quad \neg\phi \wedge \neg\psi \models \phi \circ \psi \end{cases}$$

$$f_\circ(0,\mathrm{i}) = \begin{cases} 0 & \textit{iff} \quad \neg\phi \models (\psi \vee \neg\psi) \vee \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \neg\phi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \psi \vee \neg\psi \\ 1 & \textit{iff} \quad \neg\phi \models (\psi \vee \neg\psi) \vee (\phi \circ \psi) \end{cases}$$

$$f_\circ(0,1) = \begin{cases} 0 & \textit{iff} \quad \neg\phi \wedge \psi \models \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \neg\phi \wedge \psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \chi \\ 1 & \textit{iff} \quad \neg\phi \wedge \psi \models \phi \circ \psi \end{cases}$$

$$f_\circ(\mathrm{i},0) = \begin{cases} 0 & \textit{iff} \quad \neg\psi \models (\phi \vee \neg\phi) \vee \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \neg\psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \phi \vee \neg\phi \\ 1 & \textit{iff} \quad \neg\psi \models (\phi \vee \neg\phi) \vee (\phi \circ \psi) \end{cases}$$

$$f_\circ(\mathrm{i},\mathrm{i}) = \begin{cases} 0 & \textit{iff} \quad \models (\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models (\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \\ 1 & \textit{iff} \quad \models (\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee (\phi \circ \psi) \end{cases}$$

$$f_\circ(\mathrm{i},1) = \begin{cases} 0 & \textit{iff} \quad \psi \models (\phi \vee \neg\phi) \vee \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \phi \vee \neg\phi \\ 1 & \textit{iff} \quad \psi \models (\phi \vee \neg\phi) \vee (\phi \circ \psi) \end{cases}$$

$$f_\circ(1,0) = \begin{cases} 0 & \textit{iff} \quad \phi \wedge \neg\psi \models \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \phi \wedge \neg\psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \chi \\ 1 & \textit{iff} \quad \phi \wedge \neg\psi \models \phi \circ \psi \end{cases}$$

$$f_\circ(1,\mathrm{i}) = \begin{cases} 0 & \textit{iff} \quad \phi \models (\psi \vee \neg\psi) \vee \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \phi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \psi \vee \neg\psi \\ 1 & \textit{iff} \quad \phi \models (\psi \vee \neg\psi) \vee (\phi \circ \psi) \end{cases}$$

$$f_\circ(1,1) = \begin{cases} 0 & \textit{iff} \quad \phi \wedge \psi \models \neg(\phi \circ \psi) \\ \mathrm{i} & \textit{iff} \quad \phi \wedge \psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi) \models \chi \\ 1 & \textit{iff} \quad \phi \wedge \psi \models \phi \circ \psi \end{cases}$$

The i-rules[5] are considered to be elimination rules while the remaining ones are considered to be introduction rules. $\mathfrak{ND}_{\mathbf{K_3^\circ}}$ is an extension of $\mathfrak{ND}_{\mathbf{K_3}}$ by the inference rules which are introduced in Theorem 1. Note that each of $\mathfrak{ND}_{\mathbf{K_3}}$'s extensions has one, and only one, rule from each of the nine groups of the inference rules. As follows from [55], $\mathfrak{ND}_{\mathbf{K_3^\circ}}$ is sound and complete.

Let us introduce an example of a derivation of $\neg p$ from $\neg q$ and $p \circ q$ in some $\mathfrak{ND}_{\mathbf{K_3^\circ}}$ with $R_\circ(\mathrm{i}, 0, \mathrm{i})$ and $R_\circ(1, 0, \mathrm{i})$.[6]

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption |
| (2) | $p \circ q$ | assumption |
| (3) | $(p \circ q) \vee \neg(p \circ q)$ | $(\vee \mathrm{I}_1)$: 2 |
| (4) | $p \vee \neg p$ | $R_\circ(\mathrm{i}, 0, \mathrm{i})$: 1, 3 |
| (5) | $p$ | assumption |
| (6) | $p \wedge \neg q$ | $(\wedge \mathrm{I})$: 1, 5 |
| (7) | $\neg p$ | $R_\circ(1, 0, \mathrm{i})$: 6, 3 |
| (8) | $\neg p$ | assumption |
| (9) | $\neg p$ | $(\vee \mathrm{E})$: 4, 7, 8 [5-7], [8] |

## 3. Proof search for $\mathfrak{ND}_{\mathbf{K_3}}$ and $\mathfrak{ND}_{\mathbf{K_3^\circ}}$

As far as we know, there is no published work concerning natural deduction proof search for $\mathbf{K_3}$. Here we adapt an algorithm for searching for natural deduction proofs in a variety of logics [9, 10].

Note that proof-searching is always carried out in a particular system.

In searching for a proof, there are internal states of the algorithm: *list_proof* and *list_goals*. *List_proof* is a name of a (possibly empty) sequence of formulae (some of them being highlighted as discarded, following the 2nd clause of Definition 3), and *list_goals* is a name of a nonempty sequence of goals, i.e. formulae we want to prove.[7] The result

[5] By an *x-rule* we mean an $\circ$-rule with $v(\phi \circ \psi) = x$. $R_\circ(0, 0, \mathrm{i})$ is an example of an i-rule.

[6] Note that we obtain $R_\circ(\mathrm{i}, 0, \mathrm{i})$ and $R_\circ(1, 0, \mathrm{i})$ from $f_\circ(\mathrm{i}, 0) = \mathrm{i}$ and $f_\circ(1, 0) = \mathrm{i}$, respectively.

[7] As the reader will soon find out, the emptiness of *list_proof* is a specific feature of proof-searching for $\mathbf{K_3}$. A counterexample is extracted even from empty *list_proof*.

of proof-searching may be a proof or a counterexample. Below we describe in detail all the procedures to define an *algo_derivation* for $\mathfrak{ND}_{\mathbf{K_3}}$ and $\mathfrak{ND}_{\mathbf{K_3^\circ}}$ (abbreviated through the paper as $\mathfrak{ALG}_{\mathbf{K_3}}$ and $\mathfrak{ALG}_{\mathbf{K_3^\circ}}$, respectively).

Given the task of finding an *algo_derivation* of $\alpha$ from $\Gamma$ in some $\mathfrak{ND}_{\mathbf{K_3^\circ}}$, with $\Omega$ and $\Xi$ being its *list_proof* and *list_goals*, respectively, we will use the following notation. '$\Omega \vdash \Xi$' is to mean that at the current state of proof-searching $\Omega$ denotes *list_proof* and $\Xi$ denotes *list_goals*. If we want to highlight that a formula $\psi$ is a member of $\Omega$ / a goal $\psi$ is a member of $\Xi$ we write '$\Omega_1, \psi, \Omega_2 \vdash \Xi$' / '$\Omega \vdash \Xi_1, \psi, \Xi_2$'. If we want to highlight that a goal $\psi$ is the last member of $\Xi$ we write '$\Omega \vdash \Xi_1, \psi$'.

DEFINITION 4. We say *current_goal* $\psi$ is reached iff (1) $\Omega_1, \psi, \Omega_2 \vdash \Xi, \psi$ or (2) $\Omega_1, \omega, \Omega_2, \neg\omega, \Omega_3 \vdash \Xi, \psi$.

*Current_goal* $\psi$ is reached in the following situations: (1) $\psi$ is in *list_goals* and *list_proof*; (2) some formula and its negation are in *list_proof*. In the case of (2) *current_goal* $\psi$ is reached via (EFQ).

DEFINITION 5. We say '$\Omega \vdash \Xi$' leads to '$\Omega_1 \vdash \Xi_1$', if an algorithm takes '$\Omega \vdash \Xi$' as an input and outputs with '$\Omega_1 \vdash \Xi_1$'. This definition captures the idea that at some point during a proof search the current state is '$\Omega \vdash \Xi$' and some time later it is '$\Omega_1 \vdash \Xi_1$'.

PROCEDURE 1 (Pr1). Pr1 governs the applicability of the elimination rules. Pr1 marks both its premise(s) and a conclusion to prevent the reapplicability of the same elimination rule to the same formula. On the other hand, reapplicability of the same elimination rule to the same formula becomes possible, if the conclusion of the previous application of the elimination rule is discarded from *list_proof*.

PROCEDURE 2 (Pr2). Pr2 governs the reachability of *current_goal*. Pr2 subsequently checks whether *list_proof* has *current_goal* as a non-discarded formula and (if not) checks whether *list_proof* has some formula and its negation, both being non-discarded.[8] Pr2 deletes a reached goal from *list_goals* and sets an element in *list_goals* preceding it, if existent, as *current_goal*.

PROCEDURE 3 (Pr3). Pr3 governs the updatabilty of both *list_proof* and *list_ goals* when Pr1 fails and *current_goal* isn't reached. The purpose of Pr3 is to find new goals, if needed. Pr3 consists of Pr3.1 and

[8] Let us recall that the notion of a discarded formula is defined in the 2nd clause of Definition 3.

Pr3.2. Pr3.1 governs updatabilty of *list_goals* and Pr3.2 governs the updatabilty of *list_proof*.

PROCEDURE 3.1 (Pr3.1). Pr3.1 governs *current_goal* depending on its type. Below Pr3.1.1–Pr3.1.7 provide a description in detail. If *current_goal* isn't reached, then *current_goal* is updated and *list_proof* is updated. Note that we do nothing, if *current_goal* is a literal.[9]

Pr3.1.1. If $\Omega \vdash \Xi, \phi \wedge \psi$ and $\phi \wedge \psi$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \phi \wedge \psi, \phi$ and afterwards $\Omega, \phi \vdash \Xi, \phi \wedge \psi, \psi$ (a proof of a conjunctive goal is searched via searching for proofs of its conjuncts starting from the left one).

Pr3.1.2. If $\Omega \vdash \Xi, \phi \vee \psi$ and $\phi \vee \psi$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \phi \vee \psi, \phi$ or $\Omega \vdash \Xi, \phi \vee \psi, \psi$ (a proof of a disjunctive goal is searched via searching for proofs of its disjuncts starting from the left one).[10]

Pr3.1.3. If $\Omega \vdash \Xi, \neg(\phi \wedge \psi)$ and $\neg(\phi \wedge \psi)$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \neg(\phi \wedge \psi), \neg\phi \vee \neg\psi$ (proof-searching for $\neg\phi \vee \neg\psi$ guides us to proof-searching for $\neg(\phi \wedge \psi)$).

Pr3.1.4. If $\Omega \vdash \Xi, \neg(\phi \vee \psi)$ and $\neg(\phi \vee \psi)$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \neg(\phi \vee \psi), \neg\phi \wedge \neg\psi$ (proof-searching for $\neg\phi \wedge \neg\psi$ guides us to proof-searching for $\neg(\phi \vee \psi)$).

Pr3.1.5. If $\Omega \vdash \Xi, \neg\neg\phi$ and $\neg\neg\phi$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \neg\neg\phi, \phi$ (proof-searching for $\phi$ guides us to proof-searching for $\neg\neg\phi$).

Pr3.1.6-Pr.3.1.7 govern the cases with $\phi \circ \psi$ or $\neg(\phi \circ \psi)$ being *current_goal*. The way we treat them depends solely on $\circ$-rules the particular system has.

The general idea is that only the 1-rules of the system in question (if any) are responsible for searching for *current_goal* $\phi \circ \psi$ and only the

[9] A literal is standardly defined to be a propositional variable or its negation.

[10] Note that the difference between Pr3.1.1 and Pr3.1.2. In the former procedure, the left conjunct is reached if and only if we start out searching for the right one. In the latter procedure, the left disjunct is reached if and only if we *don't* start out searching for the right one. In more detail, if the left disjunct is reached, then we don't start out searching for the right one because a disjunctive goal is reachable via $(\vee I_1)$. Conversely, if we don't start out searching for the right disjunct, then it means we have already reached the left one and, therefore, a disjunctive goal is reached via $(\vee I_1)$. Additionally, in the case of Pr3.1.2 we delete all the formulae and goals to have been added to an algo-derivation while searching for the left disjunct. It should be noted that 'deleted' means something considerably different than 'discarded'. For example, discarded formulae are one of the essential parts of a derivation while, having been one of the essential parts of proof-searching, deleted formulae are not parts of a derivation at all.

0-rules of the system in question (if any) are responsible for searching for *current_goal* $\neg(\phi \circ \psi)$. For the considerations concerning the 0-rules are similar to the ones concerning the 1-rules we confine ourselves to the 1-rules.

Among the 1-rules, the easiest rule to apply is $R_\circ(\mathrm{i},\mathrm{i},1)$: it has no premises. If it is in the system and *current_goal* is $\phi \circ \psi$, then $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee (\phi \circ \psi)$ is in *list_proof*.[11] This formula in *list_proof* is helpful in searching for $\phi \circ \psi$.

Then it is easier to apply a 1-rule, if its premise is a conjunct of some conjunctive formula and not a conjunctive formula itself. Thus the next ones to apply are the rules $R_\circ(0,\mathrm{i},1)$, $R_\circ(1,\mathrm{i},1)$, $R_\circ(\mathrm{i},1,1)$, $R_\circ(\mathrm{i},0,1)$.[12] We consider $R_\circ(0,\mathrm{i},1)$. The others are considered similarly. If $R_\circ(0,\mathrm{i},1)$ is in the system, *current_goal* is $\phi \circ \psi$ and $\neg\phi$ is in *list_proof*, then $(\psi \vee \neg\psi) \vee (\phi \circ \psi)$ is in *list_proof*. This formula in *list_proof* is, again, helpful in searching for $\phi \circ \psi$.

The hardest 1-rules to apply are $R_\circ(0,0,1)$, $R_\circ(0,1,1)$, $R_\circ(1,0,1)$, $R_\circ(1,\ 1,1)$ for their premises are conjunctive formulas. We consider $R_\circ(0,0,1)$. The others are considered similarly. If $R_\circ(0,0,1)$ is in the system, *current_goal* is $\phi \circ \psi$ and $\neg\phi \wedge \neg\psi$ is in *list_proof*, then $\phi \circ \psi$ is readily reached. We thank one of the referees for the following generalisation for these procedures: "if a rule $R$ is in the system, *current_goal* is $\phi \circ \psi$ and all formulas from the left-hand side of $\models$ in $R$ occur in *list_proof*, then we also have everything that occurs on the right-hand side of $\models$ in $R$ in *list_proof*". We, however, strongly believe that a more detailed description makes understanding easier.

So, this is the fixed order in applying the 1-rules in searching for *current_goal* $\phi \circ \psi$.[13] We try to saturate *list_proof* with (negations) of subformulae of $\phi \circ \psi$.

Pr3.1.6.1. If $\Omega \vdash \Xi$, $\phi \circ \psi$ and $\phi \circ \psi$ isn't a member of $\Omega$, then $\Omega$, $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee (\phi \circ \psi) \vdash \Xi$, $\phi \circ \psi$ (a proof of $\phi \circ \psi$ is searched for via searching for the applicability of the rule $R_\circ(\mathrm{i},\mathrm{i},1)$).

Pr3.1.6.2. If $\Omega \vdash \Xi$, $\phi \circ \psi$ and $\phi \circ \psi$ isn't a member of $\Omega$, then $\Omega \vdash \Xi$, $\phi \circ \psi$, $(\psi \vee \neg\psi) \vee (\phi \circ \psi)$, $\neg\phi$ or $\Omega \vdash \Xi$, $\phi \circ \psi$, $(\phi \vee \neg\phi) \vee (\phi \circ \psi)$, $\neg\psi$ or

[11] Note that *current_goal* determines applicability of $R_\circ(\mathrm{i},\mathrm{i},1)$.

[12] The order of their application is arbitrary. The algorithm just scans *list_proof* for all the possible applications of these rules in the same way as it scans for all occurrences of conjunctive formulae in order to apply ($\wedge\mathrm{E}_1$) and ($\wedge\mathrm{E}_2$).

[13] This order, of course, isn't necessary.

$\Omega \vdash \Xi, \phi \circ \psi, (\phi \vee \neg\phi) \vee (\phi \circ \psi), \psi$ or $\Omega \vdash \Xi, \phi \circ \psi, (\psi \vee \neg\psi) \vee (\phi \circ \psi), \phi$ (a proof of $\phi \circ \psi$ is searched for via searching for the applicability of one of the rules $R_\circ(0,\mathrm{i},1)$, $R_\circ(\mathrm{i},0,1)$, $R_\circ(\mathrm{i},1,1)$, $R_\circ(1,\mathrm{i},1)$).[14]

Pr3.1.6.3. If $\Omega \vdash \Xi, \phi\circ\psi$ and $\phi\circ\psi$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \phi\circ\psi, \neg\phi\wedge\neg\psi$ or $\Omega \vdash \Xi, \phi\circ\psi, \neg\phi\wedge\psi$ or $\Omega \vdash \Xi, \phi\circ\psi, \neg\phi\wedge\neg\psi$ or $\Omega \vdash \Xi, \phi\circ\psi, \phi\wedge\neg\psi$ or $\Omega \vdash \Xi, \phi\circ\psi, \phi\wedge\psi$ (a proof of $\phi\circ\psi$ is searched for via searching for the applicability of one of the rules $R_\circ(0,0,1)$, $R_\circ(0,1,1)$, $R_\circ(1,0,1)$, $R_\circ(1,1,1)$).[15]

Pr3.1.7.1. If $\Omega \vdash \Xi, \neg(\phi\circ\psi)$ and $\neg(\phi\circ\psi)$ isn't a member of $\Omega$, then $\Omega, (\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee \neg(\phi\circ\psi) \vdash \Xi, \neg(\phi\circ\psi)$ (a proof of $\neg(\phi\circ\psi)$ is searched for via searching for the applicability of the rule $R_\circ(\mathrm{i},\mathrm{i},0)$).

Pr3.1.7.2. If $\Omega \vdash \Xi, \neg(\phi\circ\psi)$ and $\neg(\phi\circ\psi)$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \neg(\phi\circ\psi), (\psi \vee \neg\psi) \vee \neg(\phi\circ\psi), \neg\phi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), (\phi \vee \neg\phi) \vee \neg(\phi\circ\psi), \neg\psi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), (\phi \vee \neg\phi) \vee \neg(\phi\circ\psi), \phi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), (\psi \vee \neg\psi) \vee \neg(\phi\circ\psi), \phi$ (a proof of $\neg(\phi\circ\psi)$ is searched via searching for applicability one of the rules $R_\circ(0,\mathrm{i},0)$, $R_\circ(\mathrm{i},0,0)$, $R_\circ(\mathrm{i},1,0)$, $R_\circ(1,\mathrm{i},0)$).[16]

Pr3.1.7.3. If $\Omega \vdash \Xi, \neg(\phi\circ\psi)$ and $\neg(\phi\circ\psi)$ isn't a member of $\Omega$, then $\Omega \vdash \Xi, \neg(\phi\circ\psi), \neg\phi\wedge\neg\psi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), \neg\phi\wedge\psi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), \neg\phi\wedge\psi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), \phi\wedge\neg\psi$ or $\Omega \vdash \Xi, \neg(\phi\circ\psi), \phi\wedge\psi$ (a proof of $\neg(\phi\circ\psi)$ is searched for via searching for the applicability of one of the rules $R_\circ(0,0,0)$, $R_\circ(0,1,0)$, $R_\circ(1,0,0)$, $R_\circ(1,1,0)$).[17]

Procedure 3.2. Pr3.2 governs *current_goal* depending on the type of formulae in *list_proof*. Below Pr3.2.1-Pr3.2.2 provide a detailed description. As in Pr1, Pr3.2 prevents reapplicability with marks. First, Pr3.2 doesn't apply to the formulae which are marked by Pr1. Second, Pr3.2 marks the reapplicability of the same rule to the same formula. However, it allows the applicability of two (three etc.) rules to the same formula. On the other hand, the reapplicability of Pr3.2 to the same formula becomes possible, if the result of its application is discarded from *list_proof*.

[14] Pr3.1.6.2 is performed after Pr3.1.6.1 only. Note that proof-searching is performed with deleting formulae which have appeared in *list_proof* and *list_goals* via Pr3.1.6.1. (This is the same way we deal with a disjunctive formula in Pr3.1.2.)

[15] Pr3.1.6.3 is performed after Pr3.1.6.2 only.

[16] Pr3.1.7.2 is performed after Pr3.1.7.1 only.

[17] Pr3.1.7.3 is performed after Pr3.1.7.2 only.

Pr3.2.1. If $\Omega_1, \phi \vee \psi, \Omega_2 \vdash \Xi, \chi$, then $\Omega_1, \phi \vee \psi, \Omega_2, \phi \vdash \Xi, \chi, \chi$ and afterwards $\Omega_1, \phi \vee \psi, \Omega_2, \chi, \psi \vdash \Xi, \chi, \chi$ and afterwards $\Omega_1, \phi \vee \psi, \Omega_2, \chi \vdash \Xi$ (if an unmarked formula $\phi \vee \psi$ is in *list_proof* and *current_goal* $\chi$ isn't reached, then *list_proof* and *list_goals* are updated with an assumption $\phi$ and new *current_goal* $\chi$, respectively; if $\chi$ is reached, then an assumption $\phi$ is discarded, an assumption $\psi$ is added, *current_goal* $\chi$ is deleted and new *current_goal* $\chi$ is set; if $\chi$ is reached with an assumption $\psi$ this time, then an assumption $\psi$ is discarded, ($\vee$E) is applied, both goals $\chi, \chi$ are deleted and the preceding goal is set as *current_goal*).[18]

The general idea is to use $\phi \circ \psi$ and $\neg(\phi \circ \psi)$ from *list_proof* in order to reach *current_goal*. Here the i-rules come into play. As in the case of the 1-rules and the 0-rules in Pr3.1, the i-rules are ordered by the ease of applicability. Since the considerations concerning $\phi \circ \psi$ from *list_proof* are similar to the ones concerning $\neg(\phi \circ \psi)$ from *list_proof* we confine ourselves to $\phi \circ \psi$ from *list_proof*.

Among the i-rules, the easiest rule to apply is $R_\circ(\mathrm{i}, \mathrm{i}, \mathrm{i})$: it has one premise. If it is in the system and $\phi \circ \psi$ is in *list_proof*, then *list_goals* updates with the following goals: $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi)$, $(\phi \circ \psi) \vee \neg(\phi \circ \psi)$. For $(\phi \circ \psi) \vee \neg(\phi \circ \psi)$ is readily reached via Pr3.1.2, $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi)$ is readily reached too. This formula in *list_proof* is helpful in searching for any *current_goal*.

Then it is easier to apply an i-rule, if one of its two premises is a conjunct of some conjunctive formula and not the conjunctive formula itself. Thus the next ones to apply are the rules $R_\circ(0, \mathrm{i}, \mathrm{i})$, $R_\circ(1, \mathrm{i}, \mathrm{i})$, $R_\circ(\mathrm{i}, 1, \mathrm{i})$, $R_\circ(\mathrm{i}, 0, \mathrm{i})$. We consider $R_\circ(0, \mathrm{i}, \mathrm{i})$. The others are considered similarly. If $R_\circ(0, \mathrm{i}, \mathrm{i})$ is in the system and $\phi \circ \psi$ is in *list_proof*, then *list_goals* updates with the following goals: $\psi \vee \neg\psi$, $\neg\phi$, $(\phi \circ \psi) \vee \neg(\phi \circ \psi)$. For $(\phi \circ \psi) \vee \neg(\phi \circ \psi)$ is readily reached via Pr3.1.2, $\neg\phi$ becomes *current_goal*. This goal is helpful in searching for the preceding goal.

The hardest i-rules to apply are $R_\circ(0, 0, \mathrm{i})$, $R_\circ(0, 1, \mathrm{i})$, $R_\circ(1, 0, \mathrm{i})$, $R_\circ(1, 1, \mathrm{i})$ for they have some conjunctive formulas as one of their premises. We consider $R_\circ(0, 0, \mathrm{i})$. The others are considered similarly. If $R_\circ(0, 0, \mathrm{i})$ is in the system and $\phi \circ \psi$ is in *list_proof*, then *list_goals* updates with the following goals: $\neg\phi \wedge \neg\psi$, $(\phi \circ \psi) \vee \neg(\phi \circ \psi)$. For $(\phi \circ \psi) \vee \neg(\phi \circ \psi)$ is readily reached via Pr3.1.2, $\neg\phi \wedge \neg\psi$ becomes *current_goal*. This goal is helpful in searching for the preceding goal.

[18] Note that despite new *current_goal* being the same formula as the preceding goal, these goals aren't the same goal in *list_goals*.

So, this is the order in applying the i-rules in searching for *current_goal*. Again, we try to saturate *list_proof* with (negations) of subformulae of $\phi \circ \psi$, if $\phi \circ \psi$ is in *list_proof*.

In what follows, $\varpi$ denotes $\phi \circ \psi$ or $\neg(\phi \circ \psi)$.

Pr3.2.2.1. If $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi$, then $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, (\phi \vee \neg\phi) \vee (\psi \vee \neg\psi), (\phi \circ \psi) \vee \neg(\phi \circ \psi)$.

Pr3.2.2.2. If $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi$, then $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \psi \vee \neg\psi, \neg\phi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$ and afterwards $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \phi \vee \neg\phi, \neg\psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$ and afterwards $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \phi \vee \neg\phi, \psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$ and afterwards $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \psi \vee \neg\psi, \phi\ (\phi \circ \psi) \vee \neg(\phi \circ \psi)$.

Pr3.2.2.3. If $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi$, then $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \neg\phi \wedge \neg\psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$ and afterwards $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \neg\phi \wedge \psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$ and afterwards $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \phi \wedge \neg\psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$ and afterwards $\Omega_1, \varpi, \Omega_2 \vdash \Xi, \chi, \phi \wedge \psi, (\phi \circ \psi) \vee \neg(\phi \circ \psi)$.

PROCEDURE 4. Pr4 governs the applicability of the introduction rules. It is strictly determined by and following Pr3.1 that *list_proof* is updated via an introduction rule. For example, Pr3.1.1 is formulated so that when both $\phi$ and $\psi$ are in *list_proof*, $\phi \wedge \psi$ is in *list_proof* via ($\wedge$I), either. As both Pr1 and Pr3, Pr4 prevents reapplicability with marks. Pr4 marks reapplicability to the same formula. On the other hand, the reapplicability of Pr4 to the same formula becomes possible, if the result of its application is discarded from *list_proof*. The specifics of the marking mechanism of Pr4 amount to the fact that results of its application are marked to prevent the applicability of both Pr1 and Pr3.

### Proof-searching algorithm $\mathfrak{ALG}_{\mathbf{K}_3^\circ}$

A description of the proof-searching algorithm $\mathfrak{ALG}_{\mathbf{K}_3^\circ}$ is as follows. A flowchart of the algorithm one can find in Figure 1 on page 237.

**Step 1.** Input a task to find an algo-derivation of $\alpha$ from $\Gamma$.
Goto step 2.
**Step 2.** Pr2 is launched.
The algorithm checks if *current_goal* is reached.
**if** yes **then**
    goto step 3.
**else** goto step 4.
**end if**
**Step 3.** The algorithm checks if *current_goal* is *initial_goal*.
**if** yes **then**

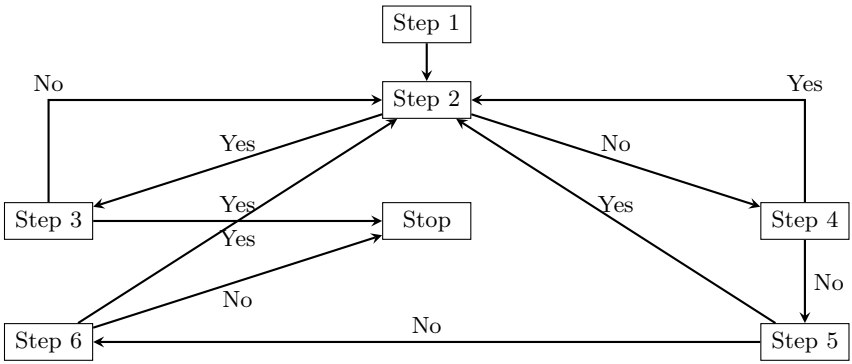


Figure 1. A flowchart of the algorithm

an algo-derivation of $\alpha$ from $\Gamma$ is found. Stop.
**else** Pr4 is launched. Goto step 2.
**end if**
**Step 4.** The algorithm checks if any elimination rule is applicable.
**if** yes **then**
Pr1 is launched. Goto step 2.
**else** goto step 5.
**end if**
**Step 5.** The algorithm checks if Pr 3.1 is applicable.
**if** yes **then**
goto step 2.
**else** goto step 6.
**end if**
**Step 6.** The algorithm checks if any unmarked formulae are in *list_proof*.
**if** yes **then**
Pr3.2 is launched. Goto step 2.
**else** a counterexample is extracted. Stop.
**end if**

## 4. Algo-Proof Examples

As an example, we consider an algo-proof of $\neg p$ from $\neg q$ and $p \circ q$ in some system with the following i-rules only: $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$, $R_\circ(1,0,\mathrm{i})$ and $R_\circ(\mathrm{i},0,\mathrm{i})$.

$\neg p$ is set to be *initial_goal* in *list_goals* with both $\neg q$ and $p \circ q$ being added in *list_proof* as assumptions. Pr1 isn't applicable, and *current_goal* isn't reached. Following the order in Pr3.2.2, Pr3.2.2.1, which governs the applicability of $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$, is of more priority than Pr3.2.2.2, which governs applicability of $R_\circ(\mathrm{i},0,\mathrm{i})$, and is of more priority than Pr3.2.2.3, which governs applicability of $R_\circ(1,0,\mathrm{i})$. By Pr3.2.2.1, $(p \vee \neg p) \vee (q \vee \neg q)$ and $(p \circ q) \vee \neg(p \circ q)$ are added to *list_goals* with $(p \circ q) \vee \neg(p \circ q)$ being *current_goal*. Pr3.1.2 applies to $(p \circ q) \vee \neg(p \circ q)$ and sets $p \circ q$ as *current_goal*. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption; *list_goals*: $\neg p$ |
| (2) | $p \circ q$ | assumption; *list_goals*: $(p \vee \neg p) \vee (q \vee \neg q)$ |
| (3) | | *list_goals*: $(p \circ q) \vee \neg(p \circ q)$ |
| (4) | | *list_goals*: $p \circ q$ |

By Pr2, *current_goal* is reached and $(p \circ q) \vee \neg(p \circ q)$ is set as *current_goal*. By Pr4, $(\vee I_1)$ applies to derive $(p \circ q) \vee \neg(p \circ q)$ in *list_proof* and it is marked. Now $(p \circ q) \vee \neg(p \circ q)$ is reached and $(p \vee \neg p) \vee (q \vee \neg q)$ is derived in *list_proof* by thus making $(p \vee \neg p) \vee (q \vee \neg q)$ reached. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption; *list_goals*: $\neg p$ |
| (2) | $p \circ q$ | assumption |
| (3) | $(p \circ q) \vee \neg(p \circ q)$ | $(\vee I_1)$: 2 |
| (4) | $(p \vee \neg p) \vee (q \vee \neg q)$ | $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$: 3 |

$(p \vee \neg p) \vee (q \vee \neg q)$ is the only unmarked formula in *list_proof*, so Pr3.2.1 applies to it. The algorithm tries to derive *current_goal* $\neg p$ from, first, the assumption $p \vee \neg p$ and then from the assumption $q \vee \neg q$. By applying Pr3.2.1 to the assumption $p \vee \neg p$ the algorithm tries to derive *current_goal* $\neg p$ from, first, the assumption $p$ and then from the assumption $\neg p$. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption; *list_goals*: $\neg p$ |
| (2) | $p \circ q$ | assumption; *list_goals*: $\neg p$ from $q \vee \neg q$ |
| (3) | $(p \circ q) \vee \neg(p \circ q)$ | $(\vee I_1)$: 2; *list_goals*: $\neg p$ from $p \vee \neg p$ |

| | | |
|---|---|---|
| (4) | $(p \vee \neg p) \vee (q \vee \neg q)$ | $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$: 3; *list_goals*: $\neg p$ from $\neg p$ |
| (5) | $p \vee \neg p$ | assumption; *list_goals*: $\neg p$ from $p$ |
| (6) | $p$ | assumption |

By Pr3.2.2.3, which governs applicability of $R_\circ(1,0,\mathrm{i})$, $p \wedge \neg q$ is set to be *current_goal*. Then Pr3.1.1 applies to it with $\neg q$, $p$ being added to *list_goals* and $p$ being *current_goal*. Because $p$ is in *list_proof*, it is reached and deleted from *list_goals*. *Current_goal* $\neg q$ is reached since it is in *list_proof*. By Pr4, $p \wedge \neg q$ is derived via ($\wedge$I) and then $R_\circ(1,0,\mathrm{i})$ applies to derive $\neg p$. A proof of $\neg p$ from $\neg p$ is trivial, so we have reached both goals $\neg p$ from $\neg p$ and $\neg p$ from $p$. So, $\neg p$ from $p \vee \neg p$ is reached too with $\neg p$ being derivable via ($\vee$E). Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption; *list_goals*: $\neg p$ |
| (2) | $p \circ q$ | assumption; *list_goals*: $\neg p$ from $q \vee \neg q$ |
| (3) | $(p \circ q) \vee \neg(p \circ q)$ | $(\vee I_1)$: 2 |
| (4) | $(p \vee \neg p) \vee (q \vee \neg q)$ | $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$: 3 |
| (5) | $p \vee \neg p$ | assumption |
| (6) | $p$ | assumption |
| (7) | $p \wedge \neg q$ | ($\wedge$I): 6, 1 |
| (8) | $\neg p$ | $R_\circ(1,0,\mathrm{i})$: 7, 3 |
| (9) | $\neg p$ | assumption |
| (10) | $\neg p$ | ($\vee$E): 9, 8, 5 [9], [6-8] |

The algorithm tries to derive *current_goal* $\neg p$ from the assumption $q \vee \neg q$. By applying Pr3.2.1 to the assumption $q \vee \neg q$, the algorithm tries to derive *current_goal* $\neg p$ from, first, the assumption $q$ and then from the assumption $\neg q$. In the first case, *current_goal* is reached, by Pr4 and via (EFQ), for both $q$ and $\neg q$ are in *list_proof*. Then the algorithm tries to derive *current_goal* $\neg p$ from the assumption $\neg q$. $p \vee \neg p$ is unmarked in *list_proof* and Pr3.2.1 applies to it. Now the algorithm, again, tries to derive *current_goal* $\neg p$ from, first, the assumption $p$ and then from the assumption $\neg p$. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption; *list_goals*: $\neg p$ |
| (2) | $p \circ q$ | assumption; *list_goals*: $\neg p$ from $q \vee \neg q$ |

| | | |
|---|---|---|
| (3) | $(p \circ q) \vee \neg(p \circ q)$ | $(\vee I_1)$: 2; *list_goals*: $\neg p$ from $\neg q$ |
| (4) | $(p \vee \neg p) \vee (q \vee \neg q)$ | $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$: 3; *list_goals*: $\neg p$ from $p \vee \neg p$ |
| (5) | $p \vee \neg p$ | assumption; *list_goals*: $\neg p$ from $\neg p$ |
| (6) | $p$ | assumption; *list_goals*: $\neg p$ from $p$ |
| (7) | $p \wedge \neg q$ | $(\wedge\mathrm{I})$: 6, 1 |
| (8) | $\neg p$ | $R_\circ(1,0,\mathrm{i})$: 7, 3 |
| (9) | $\neg p$ | assumption |
| (10) | $\neg p$ | $(\vee\mathrm{E})$: 9, 8, 5 [9], [6-8] |
| (11) | $q \vee \neg q$ | assumption |
| (12) | $q$ | assumption |
| (13) | $\neg p$ | (EFQ): 12, 1 |
| (14) | $\neg q$ | assumption |
| (15) | $p$ | assumption |

At this stage of proof-searching Pr3.2.2.3 is applicable again for the result of the previous application of this procedure (step 8) is now discarded from *list_proof*. We refer to the above passage for the detailed description. So, we have the final algo-proof. Note that *list_goals* now is empty.

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg q$ | assumption |
| (2) | $p \circ q$ | assumption |
| (3) | $(p \circ q) \vee \neg(p \circ q)$ | $(\vee I_1)$: 2 |
| (4) | $(p \vee \neg p) \vee (q \vee \neg q)$ | $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$: 3 |
| (5) | $p \vee \neg p$ | assumption |
| (6) | $p$ | assumption |
| (7) | $p \wedge \neg q$ | $(\wedge\mathrm{I})$: 6, 1 |
| (8) | $\neg p$ | $R_\circ(1,0,\mathrm{i})$: 7, 3 |
| (9) | $\neg p$ | assumption |
| (10) | $\neg p$ | $(\vee\mathrm{E})$: 9, 8, 5 [9], [6-8] |
| (11) | $q \vee \neg q$ | assumption |
| (12) | $q$ | assumption |
| (13) | $\neg p$ | (EFQ): 12, 1 |
| (14) | $\neg q$ | assumption |

| | | |
|---|---|---|
| (15) | $p$ | assumption |
| (16) | $p \wedge \neg q$ | ($\wedge$I): 15, 1 |
| (17) | $\neg p$ | $R_\circ(1, 0, \mathrm{i})$: 16, 3 |
| (18) | $\neg p$ | assumption |
| (19) | $\neg p$ | ($\vee$E): 18, 17, 5 [18], [15-17] |
| (20) | $\neg p$ | ($\vee$E): 19, 13, 11 [14-19], [12-13] |
| (21) | $\neg p$ | ($\vee$E): 20, 10, 4 [5-10], [11-20] |

Now let us introduce two more examples of an algo-proof with countermodel extracting. Let us try to prove $p \vee \neg p$.

$p \vee \neg p$ is set to be *initial_goal* in *list_goals*. By P3.1.2, $p$ is set to be *current_goal*.

Note that *list_proof* now is empty. Note also that, by Definition 3, that if *list_proof* is empty, then it is not a proof in any $\mathfrak{ND}_{\mathbf{K}_3^\circ}$.

*list_proof*

(1) *list_goals*: $p \vee \neg p$

(2) *list_goals*: $p$

Pr1 isn't applicable for *list_proof* is empty. By Pr2, $p$ isn't reached for neither $p$ is in *list_proof* nor *list_proof* contains some formula and its negation. By P3.1.2, $p$ is deleted from *list_goals* and $\neg p$ is set to be *current_goal*. Again, Pr1 isn't applicable for *list_proof* is empty. By Pr2, $\neg p$ isn't reached for neither is $\neg p$ in *list_proof* nor does *list_proof* contain some formula and its negation.

Note that *list_proof* is still empty.

*list_proof*

(1) *list_goals*: $p \vee \neg p$

(2) *list_goals*: $\neg p$

At this moment, no procedures are applicable. A counterexample is extracted from empty *list_proof* with $p \notin \Sigma$ and $\neg p \notin \Sigma$. Thus, $\xi(p) = \xi(\neg p) = \xi(p \vee \neg p) = \mathrm{i}$.

As another example, we consider an algo-proof of $p \circ q$ from $\neg p \wedge q$ in some system with the following 1-rules only: $R_\circ(1, 1, 1)$ and $R_\circ(0, \mathrm{i}, 1)$. $R_\circ(0, 1, 0)$ is a rule of the system as well.

*Initial_goal* $p \circ q$ and a formula $\neg p \wedge q$ are added to *list_goals* and *list_proof*, respectively. Pr1 (applicability of the elimination rules) applies to $\neg\phi \wedge \psi$ in order to derive $\neg p$ and $q$ via ($\wedge\mathrm{E}_1$) and ($\wedge\mathrm{E}_2$). P1 isn't

applicable, and *current_goal* isn't reached. So we proceed with Pr3.1 (analysis of *current_goal* $p \circ q$). The order in Pr3.1.6 says Pr3.1.6.2 (dealing with $R_\circ(0, \mathrm{i}, 1)$) applies before Pr3.1.6.3 (dealing with $R_\circ(1, 1, 1)$). So, $(p \vee \neg q) \vee (p \circ q)$ is set to be the *current_goal*. Formally, we have

*list_proof*

| | | | |
|---|---|---|---|
| (1) | $\neg p \wedge q$ | assumption; *list_goals*: | $p \circ q$ |
| (2) | $\neg p$ | $(\wedge \mathrm{E}_1)$: 1; | *list_goals*: $(q \vee \neg q) \vee (p \circ q)$ |
| (3) | $q$ | $(\wedge \mathrm{E}_2)$: 1 | |

Pr3.1.2 applies to *current_goal* resulting in $p \circ q$ and then $q \vee \neg q$ being added to *list_goals* with $q \vee \neg q$ being *current_goal*. Pr3.1.2 applies to it and sets $q$ as *current_goal*. By Pr2 (reachability of *current_goal*), $q$ is reached. Hence, by Pr4 (applicability of the introduction rules), $q \vee \neg q$ and then $(q \vee \neg q) \vee (p \circ q)$ are derived in *list_proof* via two applications of $(\vee I_1)$. Now $p \circ q$ is *current_goal*, again. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg p \wedge q$ | assumption; *list_goals*: $p \circ q$ |
| (2) | $\neg p$ | $(\wedge \mathrm{E}_1)$: 1 |
| (3) | $q$ | $(\wedge \mathrm{E}_2)$: 1 |
| (4) | $q \vee \neg q$ | $(\vee I_1)$: 3 |
| (5) | $(q \vee \neg q) \vee (p \circ q)$ | $(\vee I_1)$: 4 |

Note that both $q \vee \neg q$ and $(q \vee \neg q) \vee (p \circ q)$ are marked to prevent applications of Pr3.2.1 (updatability of *list_proof* and *list_goals*) to them. No procedures are applicable; however, we don't stop for the system has $R_\circ(1, 1, 1)$. Hence, we delete all the formulae from *list_proof* and *list_goals*, which have been added by Pr3.1.6.2, and launch Pr3.1.6.3. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg p \wedge q$ | assumption; *list_goals*: $p \circ q$ |
| (2) | $\neg p$ | $(\wedge \mathrm{E}_1)$: 1 |
| (3) | $q$ | $(\wedge \mathrm{E}_2)$: 1 |

$p \wedge q$ is set to be *current_goal*. Pr3.1.1 applies to it resulting in $p$ and then $q$ being added to *list_goals* with $q$ being *current_goal*. $q$ is reached, so $p$ is set to be *current_goal*. Formally, we have

*list_proof*

| | | |
|---|---|---|
| (1) | $\neg p \wedge q$ | assumption; *list_goals*: $p \circ q$ |
| (2) | $\neg p$ | ($\wedge$E$_1$): 1; *list_goals*: $\phi$ |
| (3) | $q$ | ($\wedge$E$_2$): 1 |

At this moment we stop for no procedures of the algorithm are applicable with $p \notin \Sigma$, $\neg p \in \Sigma$, $q \in \Sigma$ and $\neg q \notin \Sigma$. The following counterexample is extractable: $\xi(p) = 0$, $\xi(q) = 1$, $\xi(\neg p \wedge q) = 1$, and $\xi(p \circ q) = 0$.

## 5. Termination, soundness, and completeness

THEOREM 2 (Termination of the algorithm). *The algorithm halts on any input.*

Theorem 2 follows from two lemmata. Lemma 3 shows each procedure of the algorithm is finite. Lemma 4 shows no infinite loop is possible.

LEMMA 3. *Each procedure of the algorithm is finite.*

PROOF. We need a standard definition of the degree of a formula $\omega$, $g(\omega)$, as the number of connectives in $\omega$. Without loss of generality, we say $g(P) = g(\neg P) = 0$, for each propositional variable $P$.

We start out by showing the finiteness of Pr3.1, which is a part of Pr3 and governs an analysis of *current_goal*. With regard to Pr3.1.1–Pr3.1.5, any application of these procedures decreases, directly or indirectly, the degree of *current_goal*.

For example, Pr3.1.1 does it directly for $g(\phi) < g(\phi \wedge \psi)$ and $g(\psi) < g(\phi \wedge \psi)$. Pr3.1.4 does it indirectly. First, it applies to $\neg(\phi \vee \psi)$ to obtain *current_goal* $\neg\phi \wedge \neg\psi$. Second, Pr3.1.1 applies to $\neg\phi \wedge \neg\psi$ with $g(\neg\phi) < g(\neg(\phi \vee \psi))$ and $g(\neg\psi) < g(\neg(\phi \vee \psi))$. The reader might readily check that the same thing holds for Pr3.1.6–Pr3.1.7. By induction on a degree of *current_goal*, it can be shown that within a finite number of applications of Pr3.1 to *current_goal* the algorithm stops by obtaining a literal as *current_goal*.

The finiteness of Pr3.1 implies the finiteness of Pr4, which governs applications of the introduction rules. Note that the marking mechanism prevents reapplication of an introduction rule unless the result of this application is discarded from *list_proof*.

To show the finiteness of Pr1, which governs applications of the elimination rules, let's consider the following.

First, the number of assumptions in $\Gamma$ is finite. Second, due to the finiteness of Pr3.1, a number of assumptions added to *list_proof* via Pr3.1 is finite.

Third, it can be shown in the same manner as in the case of Pr3.1 that in any application of Pr1 (except ($\vee$E), ($\neg\wedge E$), $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$, $R_\circ(0,0,\mathrm{i})$, $R_\circ(0,1,\mathrm{i})$, $R_\circ(1,0,\mathrm{i})$, and $R_\circ(1,1,\mathrm{i})$), the degree of a conclusion is less than a degree of (some of) its premise(s).

With regard to ($\neg\wedge E$) and $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$, they are reducible to the case of ($\vee$E).

With regard to ($\vee$E), $R_\circ(0,0,\mathrm{i})$, $R_\circ(0,1,\mathrm{i})$, $R_\circ(1,0,\mathrm{i})$, and $R_\circ(1,1,\mathrm{i})$, let's recall that an arbitrary formula $\chi$, which is the conclusion of these rules, is *current_goal* (see Pr3.2). By induction, it can be shown that there can be a finite number of applications of Pr1 to a finite number of both assumptions from $\Gamma$ and assumptions added to *list_proof* via Pr3.1. Note that the marking mechanism prevents the reapplication of an elimination rule unless the result of this application is discarded from *list_proof*.

The finiteness of Pr1 implies the finiteness of Pr2, which governs reachability of *current_goal*: it is a finite process to check, if a finite *list_proof* contains *current_goal* and/or some formula and its negation.

Finally, the finiteness of both Pr1 and the number of assumptions implies the finiteness of Pr3.2, which governs *current_goal* depending on the type of formulae in *list_proof*. The same argument concerning Pr3.1 applies to Pr3.2. For example, by 3.2.1, if $\phi \vee \psi$ is in *list_proof*, then $\phi$ is in *list_proof* or $\psi$ is in *list_proof* with $g(\phi) < g(\phi \vee \psi)$ and $g(\psi) < g(\phi \vee \psi)$. By induction, it can be shown that there can be a finite number of applications of Pr3.2 to a finite *list_proof*. Note that the marking mechanism prevents the reapplication of Pr3.2 to the same formula unless the result of this application is discarded from *list_proof*. $\dashv$

LEMMA 4. *No infinite loops are possible.*

PROOF. According to the description of the algorithm, the following loops are possible.

Loop 1: Step2 $\infty$ Step3.

On Step 2, the algorithm checks the reachability of *current_goal*. If *current_goal* is reached, then the algorithm checks whether *current_goal* is *initial_goal*. If it's not, then, by Pr4, which governs applications

of introduction rules, *current_goal* is deleted from *list_goals* and the preceding goal is set to be *current_goal*. Then the algorithm returns to Step 2 to check whether new *current_goal* is *initial_goal*. Due to the finiteness of Pr3.1, which governs an analysis of *current_goal*, *list_goals* is finite. Thus, the algorithm subsequently checks each goal in *list_goals* down *initial_goal*. It implies Loop 1 is impossible.

Loop 2: Step2 $\infty$ Step4.

On Step 2, the algorithm checks the reachability of *current_goal*. If *current_goal* isn't reached, then, by Pr1, which governs applications of elimination rules, all possible elimination rules are applied in *list_proof*. Due to the finiteness of Pr1, this process stops and the algorithm returns to Step 2 to check whether *current_goal* is reached in an updated *list_proof*. If *current_goal* isn't reached, again, then Pr1 isn't applicable, and the algorithm goes to Step 5. It implies Loop 2 is impossible.

Loop 3: Step2 $\infty$ Step5.

On Step 2, the algorithm checks the reachability of *current_goal*. If *current_goal* isn't reached, then the algorithm goes to Step 4. By the previous argument, it's possible for the algorithm to go to Step 5. On this Step, the algorithm checks whether Pr3.1, which governs an analysis of *current_goal*, is applicable to *current_goal*. If it is, new *current_goal* is set and the algorithm returns to Step 2 to check whether new *current_goal* is reached. Due to the finiteness of Pr3.1, the algorithm subsequently comes down to a literal being *current_goal*. For Pr3.1 isn't applicable to a literal the algorithm goes to Step 6. It implies Loop 3 is impossible.

Loop 4: Step2 $\infty$ Step6.

On Step 2, the algorithm checks the reachability of *current_goal*. If *current_goal* isn't reached, then the algorithm goes to Step 4. By the penultimate argument, it's possible for the algorithm to go to Step 5. By the ultimate argument, it's possible for the algorithm to go to Step 6. On this step, the algorithm checks whether Pr3.2 which governs *current_goal* depending on the type of formulae in *list_proof*, is applicable. If it's applicable, then the algorithm sets new *current_goal*, marks the formula from *list_proof* to which Pr3.2 is applied and returns to Step 2. Due to the finiteness of both Pr3.2 and *list_proof*, the algorithm subsequently comes down to the situation, where each formula in *list_proof* is a literal or is marked. Since Pr3.2 isn't applicable to a literal, so the algorithm goes to a counterexample extraction. It implies Loop 4 is impossible. $\dashv$

THEOREM 5 (Soundness of the algorithm). *The algorithm is sound.*

PROOF. By Theorem 3 in [55], the system for strong Kleene logic $\mathbf{K_3}$ and the systems for each of its binary extensions are sound. Any algo-derivation is a derivation in one of the systems. Therefore, the algorithm is sound. ⊣

Note, again, that the fact that *list_proof* in an algo-proof may be empty doesn't affect the soundness of the algorithm. We just say that if there is an algo-proof of $\alpha$ from $\Gamma$ in a $\mathfrak{ND}_{\mathbf{K_3^\circ}}$, then there is a proof of $\alpha$ from $\Gamma$ in a $\mathfrak{ND}_{\mathbf{K_3^\circ}}$. To prove completeness we need the following Lemma 6. We use the idea in [10, 36].

A truth-value assignment $\xi$ of a formula $\phi$ in a model is denoted by $\xi(\varphi)$. This definition is easily extended to sets of formulae. For example, $\xi(\Gamma) = 1$ iff $\xi(\phi) = 1$ for each $\phi$ from $\Gamma$.

LEMMA 6. *If the algorithm with a task to find a derivation of $\alpha$ from (possibly, empty) $\Gamma$ in some system $\mathfrak{ND}_{\mathbf{K_3^\circ}}$ stops without finding such a derivation in this system, then list_proof contains a (possibly, empty) set $\Sigma$, $\Gamma \subseteq \Sigma$, of non-discarded formulae such that $\xi(\Gamma) = 1$ and $\xi(\alpha) \neq 1$.*

PROOF. We, first show that $\xi(\alpha) \neq 1$. A type of $\alpha$ determines the number of cases. Note that $\Sigma$ may be empty. In the text of the proof of this Lemma below, we will abbreviate the fact that there is no algo-proof of $\alpha$ from $\Gamma$ by "Condition".

1. $\alpha$ is a literal. By Condition, $\alpha \notin \Sigma$. So, $\xi(\alpha) \neq 1$.

2. $\alpha$ is $\phi \vee \psi$. By Condition and Pr3.1.2, $\phi \notin \Sigma$ and $\psi \notin \Sigma$. So, $\xi(\phi) \neq 1$ and $\xi(\psi) \neq 1$. Therefore, $\xi(\phi \vee \psi) \neq 1$.

3. $\alpha$ is $\phi \wedge \psi$. By Condition and Pr.3.1.1, $\phi \notin \Sigma$ or $\psi \notin \Sigma$. So, $\xi(\phi) \neq 1$ or $\xi(\psi) \neq 1$. Therefore, $\xi(\phi \wedge \psi) \neq 1$.

4. $\alpha$ is $\phi \circ \psi$. By Condition, $\phi \circ \psi \notin \Sigma$. In total there are sixteen outputs depending on whether $\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\neg\psi \in \Sigma$. We exclude seven outputs, where ($\phi \in \Sigma$ and $\neg\phi \in \Sigma$) or ($\psi \in \Sigma$ and $\neg\psi \in \Sigma$): otherwise, there would be a proof of $\phi \circ \psi$ from $\Gamma$, by (EFQ). So, we come up with the following nine outputs:

4.1. $\phi \in \Sigma, \neg\phi \notin \Sigma, \psi \in \Sigma, \neg\psi \notin \Sigma$;
4.2. $\phi \in \Sigma, \neg\phi \notin \Sigma, \psi \notin \Sigma, \neg\psi \in \Sigma$;
4.3. $\phi \in \Sigma, \neg\phi \notin \Sigma, \psi \notin \Sigma, \neg\psi \notin \Sigma$;
4.4. $\phi \notin \Sigma, \neg\phi \in \Sigma, \psi \in \Sigma, \neg\psi \notin \Sigma$;
4.5. $\phi \notin \Sigma, \neg\phi \in \Sigma, \psi \notin \Sigma, \neg\psi \in \Sigma$;
4.6. $\phi \notin \Sigma, \neg\phi \in \Sigma, \psi \notin \Sigma, \neg\psi \notin \Sigma$;

4.7. $\phi \notin \Sigma, \neg\phi \notin \Sigma, \psi \in \Sigma, \neg\psi \notin \Sigma$;
4.8. $\phi \notin \Sigma, \neg\phi \notin \Sigma, \psi \notin \Sigma, \neg\psi \in \Sigma$;
4.9. $\phi \notin \Sigma, \neg\phi \notin \Sigma, \psi \notin \Sigma, \neg\psi \notin \Sigma$.

First, we consider one of the four outputs, where two literals belong to $\Sigma$, say, 4.1; the others are similar. Note that we always search for an algo-proof in a particular system.

Suppose the system we are looking for an algo-proof in has $R_\circ(1,1,1)$. By Pr3.1.6.3, $\phi \wedge \psi \in \Sigma$, that contradicts Condition that $\phi \circ \psi \notin \Sigma$. Therefore, $R_\circ(1,1,1)$ isn't a rule of the system in this output. So, if $\xi(\phi) = \xi(\psi) = 1$, then $\xi(\phi \circ \psi) \neq 1$.

Second, we consider one of the four outputs, where only one of the literals belongs to $\Sigma$, say, 4.3; the others are similar.

Suppose the system we are looking for an algo-proof in has $R_\circ(1,\mathrm{i},1)$. By Pr3.1.6.2, $(\psi \vee \neg\psi) \vee (\phi \circ \psi) \in \Sigma$. By Pr3.2.1, $\psi \in \Sigma$ or $\neg\psi \in \Sigma$ or $\phi \circ \psi \in \Sigma$. The latter variant contradicts Condition and the two former ones contradict Condition of output 4.3. Therefore, $R_\circ(1,\mathrm{i},1)$ isn't a rule of the system in the case of this output. So, if $\xi(\phi) = 1$ and $\xi(\psi) = \mathrm{i}$, then $\xi(\phi \circ \psi) \neq 1$.

Third, we consider 4.9, the only output, where no literal belongs to $\Sigma$.

Suppose the system we are looking for a proof in has $R_\circ(\mathrm{i},\mathrm{i},1)$. By Pr3.1.6.1, $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee (\phi \circ \psi) \in \Sigma$. By Pr3.2.1, $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\psi \in \Sigma$ or $\phi \circ \psi \in \Sigma$. The latter variant contradicts Condition while the four former ones contradict Condition of output 4.9. Therefore, $R_\circ(\mathrm{i},\mathrm{i},1)$ isn't a rule of the system in this output. So, if $\xi(\phi) = \xi(\psi) = \mathrm{i}$, then $\xi(\phi \circ \psi) \neq 1$.

5. $\alpha$ is $\neg(\phi \circ \psi)$. We treat it analogously to Case 4.

Second, we show that $\xi(\Sigma) = 1$, that is, a set $\Sigma$ is a model set. The number of cases depends on the type of a formula in $\Sigma$.

Case 0: for any formula $\phi$, it is not the case that $\phi \in \Sigma$ and $\neg\phi \in \Sigma$.

Suppose both $\phi$ and $\neg\phi$ belong to $\Sigma$. By Pr2 and Pr4, $\alpha$ is derived, which contradicts Condition.

Case 1: If $\neg\neg\phi \in \Sigma$, then $\phi \in \Sigma$ and $\neg\phi \notin \Sigma$.

If $\neg\neg\phi \in \Sigma$, then $\phi \in \Sigma$, by Pr1. If $\neg\phi \in \Sigma$, then, by Pr2 and Pr4, $\alpha$ is derived that contradicts Condition. So, we have $\phi \in \Sigma$ and $\neg\phi \notin \Sigma$. Therefore, $\xi(\phi) = \xi(\neg\neg\phi) = 1$.

Case 2: If $\phi \wedge \psi \in \Sigma$, then $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \in \Sigma$, $\neg\psi \notin \Sigma$.

If $\phi \wedge \psi \in \Sigma$, then both $\phi \in \Sigma$ and $\psi \in \Sigma$, by Pr1. By Pr2 and Pr4, both $\neg\phi \notin \Sigma$ and $\neg\psi \notin \Sigma$. So, we have $\phi \in \Sigma$, $\psi \in \Sigma$, $\neg\phi \notin \Sigma$, and $\neg\psi \notin \Sigma$. Therefore, $\xi(\phi) = \xi(\psi) = \xi(\phi \wedge \psi) = 1$.

Case 3: If $\phi \vee \psi \in \Sigma$, then $\phi \in \Sigma$ or $\psi \in \Sigma$.

If $\phi \vee \psi \in \Sigma$, then, by Pr3.2.1, $\phi \in \Sigma$ or $\psi \in \Sigma$ or $\alpha \in \Sigma$. The latter output contradicts Condition. By Pr2 and Pr4, if $\phi \in \Sigma$ or $\psi \in \Sigma$, then the outputs with the following properties contradict Condition: ($\phi \in \Sigma$ and $\neg\phi \in \Sigma$) or ($\psi \in \Sigma$ and $\neg\psi \in \Sigma$). So we have five outputs out of the sixteen ones:

3.1 $\phi \in \Sigma, \neg\phi \notin \Sigma, \psi \in \Sigma, \neg\psi \notin \Sigma$;
3.2 $\phi \in \Sigma, \neg\phi \notin \Sigma, \psi \notin \Sigma, \neg\psi \in \Sigma$;
3.3 $\phi \in \Sigma, \neg\phi \notin \Sigma, \psi \notin \Sigma, \neg\psi \notin \Sigma$;
3.4 $\phi \notin \Sigma, \neg\phi \notin \Sigma, \psi \in \Sigma, \neg\psi \notin \Sigma$;
3.5 $\phi \notin \Sigma, \neg\phi \notin \Sigma, \psi \in \Sigma, \neg\psi \notin \Sigma$.

Therefore, $\xi(\phi) = \xi(\phi \circ \psi) = 1$ or $\xi(\psi) = \xi(\phi \circ \psi) = 1$.

Case 4: If $\phi \circ \psi \in \Sigma$, then

4.1 $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \in \Sigma$, $\neg\psi \notin \Sigma$;
4.2 $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \notin \Sigma$, $\neg\psi \notin \Sigma$;
4.3 $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \notin \Sigma$, $\neg\psi \in \Sigma$;
4.4 $\phi \notin \Sigma$, $\neg\phi \notin \Sigma$, $\psi \in \Sigma$, $\neg\psi \notin \Sigma$;
4.5 $\phi \notin \Sigma$, $\neg\phi \notin \Sigma$, $\psi \notin \Sigma$, $\neg\psi \notin \Sigma$;
4.6 $\phi \notin \Sigma$, $\neg\phi \notin \Sigma$, $\psi \notin \Sigma$, $\neg\psi \in \Sigma$;
4.7 $\phi \notin \Sigma$, $\neg\phi \in \Sigma$, $\psi \in \Sigma$, $\neg\psi \notin \Sigma$;
4.8 $\phi \notin \Sigma$, $\neg\phi \in \Sigma$, $\psi \notin \Sigma$, $\neg\psi \notin \Sigma$;
4.9 $\phi \notin \Sigma$, $\neg\phi \in \Sigma$, $\psi \notin \Sigma$, $\neg\psi \in \Sigma$.

We will use the following notation. By a $(x, y)$-*cluster*, where $x, y \in \{0, \mathrm{i}, 1\}$, we mean a set of $\circ$-rules with $\xi(\phi) = x$ and $\xi(\psi) = y$. For example, the $(0, 0)$-cluster is $\{R_\circ(0,0,0), R_\circ(0,0,\mathrm{i}), R_\circ(0,0,1)\}$. Recall that by an *x-rule* we mean an $\circ$-rule with $\xi(\phi \circ \psi) = x$. $R_\circ(0,0,\mathrm{i})$ is an example of an i-rule.

We divide subcases into groups and prove this case for an arbitrary cluster from each group depending on the type of a 0-rule in it. Group 1 consists of the $(0,0)$-cluster, the $(0,1)$-cluster, the $(1,0)$-cluster and the $(1,1)$-cluster. Group 2 consists of the $(0,\mathrm{i})$-cluster, the $(1,\mathrm{i})$-cluster, the $(\mathrm{i},0)$-cluster and the $(\mathrm{i},1)$-cluster. The $(\mathrm{i},\mathrm{i})$-cluster forms group 3.

We start with group 1 and choose the $(0,0)$-cluster. The analogous argument holds, if we choose the $(0,1)$-cluster or the $(1,0)$-cluster or the $(1,1)$-cluster.

4.1. A system has $R_\circ(0,0,0)$. The Condition implies (4.1.1) $\neg\phi \notin \Gamma$ or (4.1.2) $\neg\psi \notin \Gamma$. We consider (4.1.1). The other one is symmetrical. Given $\neg\phi \notin \Gamma$, there are two outputs depending on whether $\phi \in \Gamma$. We consider (4.1.1.1), where $\phi \in \Gamma$. (4.1.1.2.) is symmetrical.

We have both $\neg\phi \notin \Gamma$ and $\phi \in \Gamma$. Let's consider the $(1,\mathrm{i})$-cluster. By correspondence analysis, the system has only one rule from this cluster.

Suppose it has $R_\circ(1,\mathrm{i},0)$. Then $(\psi \vee \neg\psi) \vee \neg(\phi \circ \psi) \in \Sigma$, by Pr3.1.7.2. By Pr3.2.1, $\psi \in \Sigma$ or $\neg\psi \in \Sigma$ or $\neg(\phi \circ \psi) \in \Sigma$. $\neg(\phi \circ \psi) \in \Sigma$ contradicts Condition. We consider first $\psi \in \Sigma$. Note that $\psi \in \Sigma$ implies $\neg\psi \notin \Sigma$, by Condition.

In the case $\phi \circ \psi \in \Sigma$, $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \in \Sigma$, $\neg\psi \notin \Sigma$ let's consider the $(1,1)$-cluster. Again, by correspondence analysis, the system has only one rule from this cluster. The system hasn't $R_\circ(1,1,0)$ for $\phi \in \Sigma$ and $\psi \in \Sigma$ imply $\neg(\phi \circ \psi) \in \Sigma$, by Pr3.1.7.3, that contradicts Condition. The system hasn't $R_\circ(1,1,\mathrm{i})$, either, for $\phi \circ \psi \in \Sigma$ implies $\phi \wedge \psi \in \Sigma$ and $(\phi \circ \psi) \vee \neg(\phi \circ \psi) \in \Sigma$, by Pr3.2.2.3. Then $\phi \wedge \psi \in \Gamma$ and $(\phi \circ \psi) \vee \neg(\phi \circ \psi) \in \Sigma$ imply $\alpha$ that contradicts Condition. Hence, if the system has $R_\circ(0,0,0)$ and $R_\circ(1,\mathrm{i},0)$, then it follows that it has $R_\circ(1,1,1)$ with $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \in \Sigma$, $\neg\psi \notin \Sigma$.

Now we consider $\neg\psi \in \Sigma$. Note that $\neg\psi \in \Sigma$ implies $\psi \notin \Sigma$, by Condition.

The case of $\phi \circ \psi \in \Sigma$, $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \notin \Sigma$, $\neg\psi \in \Sigma$ is treated analogously to the previous one by considering the $(1,0)$-cluster. We go right to Conclusion that if the system has $R_\circ(0,0,0)$ and $R_\circ(1,\mathrm{i},\mathrm{i})$, then it follows that it has $R_\circ(1,0,1)$ with $\phi \in \Sigma$, $\neg\phi \notin \Sigma$, $\psi \notin \Sigma$, $\neg\psi \in \Sigma$.

To sum up we show that if the system has $R_\circ(1,\mathrm{i},0)$, then it has $R_\circ(1,1,1)$ or $R_\circ(1,0,1)$ with $\xi(\phi \circ \psi) = 1$.

Now we suppose the system has $R_\circ(1,\mathrm{i},\mathrm{i})$. Then $(\phi \circ \psi) \vee \neg(\phi \circ \psi) \in \Sigma$, by Pr3.1.2.2 or Pr3.2.2.1. Hence, $\psi \vee \neg\psi \in \Sigma$. So, $\psi \in \Sigma$ or $\neg\psi \in \Sigma$, by Pr3.2.1. Both cases are treated above. Now we show if the system has $R_\circ(1,\mathrm{i},\mathrm{i})$, then it also has $R_\circ(1,1,1)$ or $R_\circ(1,0,1)$ with $\xi(\phi \circ \psi) = 1$.

The last rule in the $(1,\mathrm{i})$-cluster we haven't treated so far is $R_\circ(1,\mathrm{i},1)$. Then $(\phi \circ \psi) \vee \neg(\phi \circ \psi) \in \Sigma$. This case is, again, treated above.

To finally sum up case (4.1.1.1), where $\phi \in \Sigma$, we show that if the system has $R_\circ(0,0,0)$, then $\xi(\phi \circ \psi) = 1$ independently of which rule from the $(1,\mathrm{i})$-cluster the system has.

Case (4.1.1.2), where $\phi \notin \Sigma$, is treated symmetrically. So is case (4.1.2), where $\neg\psi \notin \Sigma$.

Now we finish considering case (4.1), where the system has $R_\circ(0,0,0)$.

4.2. A system has $R_\circ(0,0,\mathrm{i})$.

$\phi \circ \psi \in \Sigma$ implies $(\phi \circ \psi) \vee \neg(\phi \circ \psi) \in \Sigma$, by Pr3.2.2.3. On the other hand, Condition implies (4.2.1) $\neg\phi \notin \Gamma$ or (4.2.2) $\neg\psi \notin \Gamma$.

Both (4.2.1) and (4.2.2) are treated analogously to (4.1.1) and (4.1.2).

4.3. A system has $R_\circ(0,0,1)$. Let's consider the $(\mathrm{i},\mathrm{i})$-cluster. By correspondence analysis, the system has only one rule from this cluster.

4.3.1. The system has $R_\circ(\mathrm{i},\mathrm{i},0)$. By Pr3.1.7.1, $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee \neg(\phi \circ \psi) \in \Sigma$. By Pr3.2.1, $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\psi \in \Sigma$ or $\neg(\phi \circ \psi) \in \Sigma$. The latter output contradicts Condition.

The output $\phi \in \Sigma$ is treated above. The remaining ones are treated analogously.

4.3.2. The system has $R_\circ(\mathrm{i},\mathrm{i},\mathrm{i})$. By Pr3.2.2.1, $(\phi \circ \psi) \vee \neg(\phi \circ \psi) \in \Sigma$. Hence, $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \in \Sigma$. Pr3.2.1, $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\psi \in \Sigma$. We treat each of them as in (4.3.1).

4.3.3. The system has $R_\circ(\mathrm{i},\mathrm{i},1)$. By Pr3.1.6.1, $(\phi \vee \neg\phi) \vee (\psi \vee \neg\psi) \vee (\phi \circ \psi) \in \Sigma$. By P3.2, $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\psi \in \Sigma$ or $\phi \circ \psi \in \Sigma$. The latter output takes a little bit more detail; the others are treated above.

If $\phi \circ \psi \in \Sigma$, then we consider the other eight clusters, depending on the rules of corresponding analysis in the system in question. In this way, we either derive one of the literals $\phi, \psi$ (i.e. $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\psi \in \Sigma$) or we don't derive any of the literals $\phi, \psi$ (i.e. $\phi \notin \Sigma$ and $\neg\phi \notin \Sigma$ and $\psi \notin \Sigma$ and $\neg\psi \notin \Sigma$).

The outputs $\phi \in \Sigma$ or $\neg\phi \in \Sigma$ or $\psi \in \Sigma$ or $\neg\psi \in \Sigma$ are treated above. With regard to the output $\phi \notin \Sigma$ and $\neg\phi \notin \Sigma$ and $\psi \notin \Sigma$ and $\neg\psi \notin \Sigma$, we have $\xi(\phi) = \xi(\psi) = \mathrm{i}$ and $\xi(\phi \circ \psi) = 1$.

Now we finish treating the (0, 0)-cluster, which is a part of group 1.

Group 2 and 3 are treated similarly.

Now a proof of Case 4 is complete.

Case 5: if $\neg(\phi \wedge \psi) \in \Sigma$, then $\neg\phi \vee \neg\psi \in \Sigma$.
Case 6: if $\neg(\phi \vee \psi) \in \Sigma$, then $\neg\phi \wedge \neg\psi \in \Sigma$.
Case 7: if $\neg(\phi \circ \psi) \in \Sigma$, then there are 9 cases as in Case 4.

Proofs of cases 5 and 6 are obvious. The proof of Case 7 is analogous to the proof of Case 4. $\dashv$

The contraposition of Lemma 6 yields us a proof of the following theorem.

THEOREM 7 (Completeness of the algorithm). *The algorithm is complete.*

## 6. Related work

We begin by stressing the fact that both analytic tableaux and resolution are out of our scope due to their well-known worst-case complexity. To put it differently, these proof systems have so-called hard examples. In particular, M. D'Agostino [12] proves that analytic tableaux has factorial complexity and A. Haken [16] proves exponential complexity for resolution. This is the reason we don't discuss provers based upon these proof systems (see, for example, [6, 23]).[19]

We, also, note that it's still an open problem whether Hilbert-style calculus, sequent-style calculus with cut, and natural deduction calculus have hard examples. In addition, we avoid an influential paradigm, according to which a proof search is carried out in some proof system and then its result is transformed into natural deduction proofs [1].

On the other hand, our extensions of strong Kleene logic are limited to the 3-valued case. Therefore, in the paper we discuss neither some special case with $n \geqslant 4$ (for example, $n = 16$ as in [54]), nor the general $n$-valued case as in [18]. We consider these to be topics for future work

(Automated) proof-searching for (Jaśkowski-Fitch style) natural deduction has been the focus of much fruitful research since the nineties. Without pretending to give a full outline, let's us mention the following provers for classical logic: THINKER by J. Pelletier [32], OSCAR by J. Pollock [35], ANDP by D. Li [27], CMU PT by W. Sieg and J. Byrnes [47], and Symlog by F. Portoraro [40]. Some of them have been extended to non-classical logics and/or are accompanied with metatheoretical arguments (soundness and completeness). We note that none of them deals with the logics to have been studied in this paper and none of them provides a one-go proof-searching account (accompanied with a metatheory) for such a number of logics.

With regard to implementation, we highlight the fact that the general feature of our approach is that we propose a wide open platform for

[19] To be sure, it doesn't imply these proof systems aren't in the center of automated deduction. For the details see an influential volume [4], where natural deduction is mentioned casually.

possible implementations. It's clear that the proof-searching procedure presented in the paper needs auxiliary and specific subprocedures, if one wants to implement them for a particular logic or logics. Some derivable rules will definitely make proof-searching more effective and easier. So, these new rules will need original proof-searching procedures. With regard to first-order variants, we note that the presented proof-searching procedure has a classical first-order extension [7]. And we believe both implementation and first-order extensions for the presented logics will be a part of future research.

## 7. Conclusion

In the paper, we have presented an original finite, sound, and complete proof-searching algorithm for all the natural deduction systems for the binary extensions of strong Kleene logic. We leave for future work the task of providing the similar work for all the unary truth-functional extensions of strong Kleene logic to have been presented by Tamminga [55]. The study of complexity and proof-searching procedures for the given natural deduction systems are another points of future research.

**Acknowledgements.** The authors are very greatful to two anonymous referees for their important comments on the previous draft of the paper. Many warmly thanks go to Matthew Carmody, the linguistic editor of *LLP*, who corrected the English.

## References

[1] Andrews, P., "Classical type theory", pages 967–1007 in *Handbook of Automated Reasoning*, Elsevier Science Publishers BV, 2001. DOI: 10.1016/B978-044450813-3/50017-5

[2] Asenjo, F. G., "A calculus of antinomies", *Notre Dame Journal of Formal Logic* 7 (1966): 103–105. DOI: 10.1305/ndjfl/1093958482

[3] Avron, A., "Natural 3-valued logics — characterization and proof theory", *The Journal of Symbolic Logic* 61, 1 (1991): 276–294. DOI: 10.2307/2274919

[4] Baaz, M., C. G. Fermüller, and G. Salzer, "Automated deduction for many-valued logics", *Handbook of Automated Reasoning*, Elsevier Science Publishers BV, 2001.

[5] Batens, D., "Paraconsistent extensional propositional logics", *Logique et Analyse* 23 (90–91) (1980): 195–234.

[6] Beckert B., Hähnle R., P. Oel, and M. Sulzmann, "The tableau-based theorem prover ${}_3T^AP$ Version 4.0", pages 303–307 in M. A. McRobbie, J. K. Slaney (eds.) *Automated Deduction — Cade-13*, CADE 1996, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), 1104, 1996. DOI: 10.1007/3-540-61511-3_95

[7] Bocharov V. A., A. E. Bolotov, A. E. Gorchakov, and V. O. Shangin, "Automated first order natural deduction", pages 1292–1311 in *Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, Puna, India, 2005.

[8] Bochvar, D. A., "Ob odnom trehznachnom ischislenii i ego primenenii k analizu paradoksov klassicheskogo rasshirennogo funkcional'nogo ischislenija" (in Russian), *Sbornik: Mathematics* 4, 2 (1938): 287–308. English translation: D. A. Bochvar, "On a three-valued logical calculus and its application to the analysis of the paradoxes of the classical extended functional calculus", *History and Philosophy of Logic* 2 (1981): 87–112. DOI: 10.1080/01445348108837023

[9] Bolotov, A., A. Basukoski, O. Grigoriev, and V. Shangin, "Natural deduction calculus for linear-time temporal logic", *Lecture Notes in Computer Science* 4160 (2006): 56–68. DOI: 10.1007/11853886_7

[10] Bolotov, A., and V. Shangin, "Natural deduction system in paraconsistent setting: Proof search for PCont", *Journal of Intelligent Systems* 21 (2012): 1–24. DOI: 10.1515/jisys-2011-0021

[11] Copi, I. M., C. Cohen, and K. McMahon, *Introduction to Logic*, Fourteenth Edition, Routledge, New York, 2011.

[12] D'Agostino, M., "Are tableaux an improvement on truth-tables? Cut-free proofs and bivalence", *Journal of Logic, Language and Information* 1 (1992): 235–252. DOI: 10.1007/BF00156916

[13] Fitting, M. *First-Order Logic and Automated Theorem Proving*, Springer-Verlag, New York, 1996. DOI: 10.1007/978-1-4684-0357-2

[14] Gabbay, D. M., "What is a logical system?", pages 179–216 in D. M. Gabbay (ed.), *What Is a Logical System?*, Clarendon Press, Oxford.

[15] Gödel, K., "Zum intuitionistischen Aussgenkalkül", *Anzeiger der Akademie der Wissenschaften in Wien*, 69 (1932): 65–66. English translation: "On the intuitionistic propositional calculus", pages 300–301 in K. Gödel, *Collected Works*, Vol. 1., New York, 1986.

[16] Haken, A., "The intractability of resolution", *Theoretical Computer Science* 39 (1985): 297–308. DOI: 10.1016/0304-3975(85)90144-6

[17] Hazen, A., and F. J. Pelletier, "Gentzen and Jaśkowski natural deduction: Fundamentally similar but importantly different", *Studia Logica* 102 (2014): 1103–1142. DOI: 10.1007/s11225-014-9564-1

[18] Hähnle, R., "Automated theorem proving in multiple-valued logics", *Proc. ISMIS*, vol. 93, 1993.

[19] Heyting, A., "Die Formalen Regeln der intuitionistischen Logik. Sitzungsberichte der Preussischen Academie der Wissenschaften zu Berlin", Berlin, 1930: 42-46. English translation: "The formal rules of intuitionistic logic", pages 311-328 in P. Mancosu (ed.), *From Brouwer to Hilbert. The Debate on the Foundations of Mathematics in the 1920s*, Oxford, 1998.

[20] Indrzejczak A., "Introduction", *Studia Logica* 102 (2014): 1091–1094. DOI: 10.1007/s11225-014-9560-5

[21] Jaśkowski, S., "Recherches sur le système de la logique intuitioniste", *Actes du Congrès International de Philosophie Scientifique* 6 (1936): 58–61. English translation: "Investigations into the system of intuitionistic logic", *Studia Logica* 34, 2 (1975): 117–120.

[22] Karpenko, A., and N. Tomova, "Bochvar's three-valued logic and literal paralogics: Their lattice and functional equivalence", *Logic and Logical Philosophy* 26 (2017): 207–235. DOI: 10.12775/LLP.2016.029

[23] Kerber M., and M. Kohlhase, "A mechanization of strong Kleene logic for partial functions", pages 371–385 in A. Bundy (ed.), *Automated Deduction – CADE-12. CADE 1994*, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), 814, 1994. DOI: 10.1007/3-540-58156-1_26

[24] Kleene, S. C., *Introduction to Metamathematics*, D. Van Nostrand Company, Inc., New York, Toronto. 1952.

[25] Kleene, S. C., "On a notation for ordinal numbers", *The Journal of Symbolic Logic* 3 (1938): 150–155. DOI: 10.2307/2267778

[26] Kooi, B., and A. Tamminga, "Completeness via correspondence for extensions of the logic of paradox", *The Review of Symbolic Logic* 5 (2012): 720–730. DOI: 10.1017/S1755020312000196

[27] Li, D., "Unification algorithms for eliminating and introducing quantifiers in natural deduction automated theorem proving", *Journal of Automated Reasoning* 18 (1997): 105–134. DOI: 10.1023/A:1005749401809

[28] Łukasiewicz, J., "O logice trójwartościowej", *Ruch Filozoficzny* 5 (1920): 170–171. English translation: "On three-valued logic", pages 87-88 in L. Borkowski (ed.), *Jan łukasiewicz: Selected Works*, Amsterdam, North-Holland Publishing Company, 1997.

[29] Marcos, J., "On a problem of da Costa", pages 53–69 in *Essays of the Foundations of Mathematics and Logic*, Polimetrica International Scientific Publisher, Monza, Italy, 2005.

[30] McKinsey, J. C. C., "On the generation of the functions $C_pq$ and $N_p$ of Łukasiewicz and Tarski by means of the single binary operation", *Bulletin of the American Mathematical Society* 42 (1936): 849–851. DOI: 10.1090/S0002-9904-1936-06440-2

[31] Monteiro, A. "Construction des algèbres de Łukasiewicz trivalentes dans les algèbres de Boole monadiques. I", *Mathematica Japonica* 12 (1967): 1–23.

[32] Pelletier, F. J., "Automated natural deduction in Thinker", *Studia Logica* 60 (1998): 3–43. DOI: 10.1023/A:1005035316026

[33] Petrukhin, Y. I., "Correspondence analysis for first degree entailment", *Logical Investigations* 22, 1 (2016): 108–124.

[34] Petrukhin, Y. I., "Natural deduction system for three-valued Heyting's logic", *Moscow University Mathematics Bulletin* 72, 3 (2017): 63–66. DOI: 10.3103/S002713221703007X

[35] Pollock, J., "Natural deduction", an unpublished manuscript is available at http://johnpollock.us/ftp/OSCAR-web-page/PAPERS/Natural-Deduction.pdf

[36] Petrukhin, Y., and V. Shangin, "Automated correspondence analysis for the binary extensions of the logic of paradox", *The Review of Symbolic Logic* 10, 4 (2017): 756–781. DOI: 10.1017/S1755020317000156

[37] Petrukhin, Y., and V. Shangin, "Natural three-valued logics characterised by natural deduction", *Logique et Analyse*, accepted.

[38] Popov, V. M., "Between the logic Par and the set of all formulae" (in Russian), pages 93–95 in *The Proceeding of the 6th Smirnov Readings in logic*, Contemporary notebooks, Moscow, 2009. http://smirnovreadings.ru/en/archive/7/

[39] Popov, V. M., "On a three-valued paracomplete logic" (in Russian), *Logical Investigations* 9 (2002): 175–178. https://iphras.ru/uplfile/logic/log09/LI9_Popov.pdf

[40] Portoraro, F. "Symlog automated advice in Fitch-style proof construction", pages 802–806 in A. Bundy (ed.) *Automated Deduction — CADE-12. CADE 1994*, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), 814, 1994. DOI: 10.1007/3-540-58156-1_64

[41] Priest, G., "Paraconsistent logic", in M. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, vol. 6, Second Edition, Dordrecht, Kluwer, 2002. DOI: 10.1007/978-94-017-0460-1_4

[42] Priest, G., "The logic of paradox", *Journal of Philosophical Logic* 8 (1979): 219–241. DOI: 10.1007/BF00258428

[43] Rescher, N., *Many-Valued Logic*, New York, McGraw Hill, 1969.

[44] Sahlqvist, H., "Completeness and correspondence in the first and second order semantics for modal logic", pages 110–143 in S. Kanger (ed.), *Proceeding of the Third Scandinavian Logic Symposium*, Amsterdam, North-Holland Publishing Company, 1975.

[45] Sette, A. M., "On propositional calculus $P_1$", *Mathematica Japonica* 18 (1973): 173–180.

[46] Sette, A. M., and W. A. Carnielli, "Maximal weakly-intuitionistic logics", *Studia Logica* 55 (1995): 181–203. DOI: 10.1007/BF01053037

[47] Sieg, W. and J. Byrnes, "Normal natural deduction proofs (in classical logic)", *Studia Logica* 60 (1998): 67–106. DOI: 10.1023/A:1005091418752

[48] Shangin, V. O., "A precise definition of an inference (by the example of natural deduction systems for logics $I_{\langle\alpha,\beta\rangle}$)", *Logical Investigations* 23, 1 (2017): 83–104. DOI: 10.21146/2074-1472-2017-23-1-83-104

[49] Shestakov, V. I., "Modelling operations of propositional calculus through the relay contact circuit" (in Russian), in E. Y. Kolman, G. N. Povarov, P. V. Tavanets, and S. A. Yanovskaya (eds.), *Logical investigations*, 1959.

[50] Shestakov, V. I., "On the relationship between certain three-valued logical calculi" (in Russian), *Uspekhi Mat. Nauk* 19, 2, 116 (1964): 177–181 (available at http://www.mathnet.ru/links/573d2c9a26538eb817452537a0e26733/rm6197.pdf).

[51] Shestakov, V. I., "On one fragment of D. A. Bochvar's calculus" (in Russian), *Information Issues of Semiotics, Linguistics and Automatic Translation* VINITI, 1 (1971): 102–115.

[52] Sieg, W., and F. Pfenning, "Note by the guest editors", *Studia Logica* 60 (1998): 1. DOI: 10.1023/A:1005065031956

[53] Słupecki J., G. Bryll, and T. Prucnal, "Some remarks on the three-valued logic of J. Łukasiewicz", *Studia Logica* 21 (1967): 45–70. DOI: 10.1007/BF02123418

[54] Steen, A., and C. Benzmüller., "Sweet SIXTEEN: Automation via embedding into classical higher-order logic", *Logic and Logical Philosophy* 25, 4 (2016): 535–554. DOI: 10.12775/LLP.2016.021

[55] Tamminga, A., "Correspondence analysis for strong three-valued logic", *Logical Investigations* 20 (2014): 255–268.

[56] Tomova, N. E., "A lattice of implicative extensions of regular Kleene's logics", *Report on Mathematical Logic* 47 (2012): 173–182. DOI: 10.4467/20842589RM.12.008.0689

[57] Tomova, N. E., "Erratum to: Natural implication and modus ponens principle", *Logical Investigations* 21, 2 (2015): 186–187.

[58] Tomova, N. E., "Natural implication and modus ponens principle", *Logical Investigations* 21, 1 (2015): 138–143.

[59] van Benthem, J., "Modal correspondence theory", PhD Thesis, Universiteit van Amsterdam, Amsterdam, 1976.

[60] van Benthem, J., "Correspondence theory", pages 325–408 in D. M. Gabbay and F. Guenthner (eds), *Handbook of Philosophical Logic*, vol. 3, second edition, Dordrecht, Kluwer Academic Publishers, 2001. DOI: 10.1007/978-94-017-0454-0

Yaroslav Petrukhin and Vasilyi Shangin
Department of Logic
Lomonosov Moscow State University
Moscow, Russia
yaroslav.petrukhin@mail.ru, shangin@philos.msu.ru