Dorota Leszczyńska-Jasion
Mateusz Ignaszak
Szymon Chlebowski

Rasiowa–Sikorski Deduction Systems with the Rule of Cut: A Case Study

Abstract. This paper presents Rasiowa–Sikorski deduction systems (R–S systems) for logics CPL, CLuN, CLuNs and mbC. For each of the logics two systems are developed: an R–S system that can be supplemented with admissible cut rule, and a **KE**-version of R–S system in which the non-admissible rule of cut is the only branching rule. The systems are presented in a Smullyan-like uniform notation, extended and adjusted to the aims of this paper. Completeness is proved by the use of abstract refutability properties which are dual to consistency properties used by Fitting. Also the notion of admissibility of a rule in an R–S-system is analysed.

Keywords: R–S system, The rule of cut, Paraconsistent logics, **KE** tableau system, Inferential erotetic logic.

1. Introduction

This paper presents Rasiowa–Sikorski deduction systems ([32,33], "R–S systems" for short) for the propositional part of paraconsistent logic CLuN [2] and its two extensions: paraconsistent CLuNs [4] and logic mbC, which is a Logic of Formal Inconsistency [10]. We start the presentation with the classical case and analyse also the classical variant with equivalence. For each of the analysed logics two systems are presented. The first one is a rather standard R–S system which can be supplemented with the rule of cut to search for shorter proofs. The rule of cut is admissible in this case. The second is a version of R–S system inspired by a tableau system called "**KE**" [16,17], in which the rule of cut is the only branching rule. In this version cut is not admissible.

To our best knowledge there are no Rasiowa–Sikorski formalizations of the logics CLuN, CLuNs, mbC. In [6,7] the Reader may find tableau methods for CLuN. The Logics of Formal Inconsistency have various proof-theoretical descriptions—there is a tableau method for mbC [10], the **KE** tableau

Presented by **Andrzej Indrzejczak**; *Received* July 26, 2017

https://doi.org/10.1007/s11225-018-9795-7

method, which is also implemented [29], and there is also a sequent system for mbC [15]. In [14] the authors have also introduced resolution systems for the three logics, which are also grounded in Inferential Erotetic Logic (see Section 6 for more information concerning the logic of questions and its connection with the proof methods analysed in [14] and in this paper). We do not know, however, if there is any description of CLuN and CLuNs in terms of **KE**-like tableaux.

In this paper we also propose an extension of the uniform notation by Smullyan [34] in order to account for semantical cases which do not follow under the α-, β-scheme. The extended version allows for a uniform treatment of the analysed logics by the use of only four schemas of rules. Finally, we develop a special technique of proving completeness of the presented system—a technique using abstract *refutability properties* [12] which are dual to consistency properties proposed by Smullyan and used extensively, *e.g.*, by Fitting [18,19]. We also introduce the notion of admissibility of a rule in an R–S system and sketch some results in this field.

1.1. Motivation for This Work

Our motivation is basically proof-theoretical. We are interested in (a) relations between different deduction systems, (b) duality of proof-procedures, (c) relationship between efficiency of proof-procedures and the rule of cut. The choice of logics was motivated by our previous research presented in [14], but also by the fact that the logics we have chosen allow for a neat generalisation—by the use of an extended uniform notation the various logics are characterized by only four schemas of rules. What is more, the general unifying treatment extends to the proofs of soundness and completeness.

1.2. Logics CLuN, CLuNs, mbC

Let us now present the main characters. Logic CLuN, introduced in [2], is a predicative paraconsistent logic, and the weakest negation-complete extension of positive classical logic. CLuNs is a very rich extension of CLuN which remains paraconsistent although its axioms allow for introduction of the paraconsistent negation inside formulas (see [4]). Both logics are known from their role in the construction of inconsistency-adaptive logics (see, *e.g.*, [3]).

In [14,40] the authors have considered the propositional fragments of logics CLuN and CLuNs expressed, however, in a linguistic extension of the original logics, containing both paraconsistent and classical negation. We shall follow this approach in this paper. As in [14,40], for the sake of simplicity, we will use the names CLuN and CLuNs for the propositional logics considered here.

The Logics of Formal Inconsistency, of which mbC is the basic—minimal—example, are paraconsistent logics which "internalize" the property of consistency expressing it by the use of an operator '$\circ$'. In this way the logics recover all of Classical Logic inside the systems (see [10]). Logic mbC has been introduced in [11] and has gained popularity since then. It may be also thought of as an extension of CLuN (see [4,5]).

1.3. Invertible Rules and Confluent Systems

It is a good practice to construct deduction systems with invertible rules. One good reason for this is that such deduction systems are usually *confluent*, which means that whenever one starts with a provable formula, there are no "bad moves" in the construction of a derivation (a tree) that would lead to a "dead end", from where no proof can be found.[1]

Thus confluency seems a desirable property, favourable for implementation. However, in order to obtain invertible rules for CLuN, CLuNs and mbC we had to express them in a language richer than the object-level language of the given logic. The idea is to think of '$\sim A$' as a disjunction of a classically negated formula '$\neg A$' and a semantically atomic expression '$\chi \sim A$', where 'χ' is introduced to the language in order to express syntactically the fact that a paraconsistently negated formula '$\sim A$' can get a direct assignment of a logical value, and thus can be true even though A is also true. The idea comes from [5], and have been used successfully in [40] and later in [14], where the consistency operator '$\circ$' is treated in a similar manner.[2] The uniform notation is introduced for the richer language with 'χ'. It turns out that only in the case of CPL one language is sufficient.

1.4. Rasiowa–Sikorski Deduction Systems

In their joint paper [32], and later in the monograph [33] the authors—Helena Rasiowa and Roman Sikorski—presented the method of *diagrams*

[1] [23, p. 121]: "A tableau calculus is *proof confluent*, if from every tableau for an unsatisfiable set of sentences a closed tableau can be constructed."

[2] Let us also observe at the margin that the general idea—inspired by Suszko's Thesis—to capture a non-classical logic in a classic-like metalanguage is highly productive (see [1]), especially in proof-theory, see [9] and the discussion presented there. In [9] this idea is used to introduce cut-based analytic tableaux inspired by **KE**, just like in our account.

of formulas, which nowadays is called *Rasiowa–Sikorski method*, *Rasiowa–Sikorski diagrams*, or simply *R–S system*. Roughly speaking, a diagram of a formula is a tree with finite sequences of formulas in the nodes. The rules of the system decompose formulas in a way which is characteristic of uniform (α, β) notation. When the diagram of a formula A is completed its leaves contain finite sequences of literals. Each such leaf corresponds to a clause (disjunction) and a conjunction of such clauses is a Conjunctive Normal Form of A.

Originally, a diagram of a formula has been defined as a (partial) function taking as arguments finite binary sequences. The binary sequences "expand" step by step and "pick up" sequences of formulas as arguments. It is a beautiful definition. Unfortunately, it is out of the commonly accepted proof-theoretical tradition which leaves us with two predominant formats of proof: that of sequences and that of trees. Therefore we define the diagrams as labelled trees.

Let us also make the following useful distinction between *S-formulation* of tableau system, where single formulas occur in the nodes, and *H-formulation* of tableau system, where sets, or certain structures containing formulas, like sequences, occur in the nodes. "S" is for Smullyan, and "H" for Hintikka, and, obviously, an R–S system is an H-formulation of tableaux.

The R–S systems have been developed for many logics and found various important applications, *int.al.*, in the area of computer science—see, *e.g.*, [20,21,25,26,30]. In [20,21,30] the authors have introduced the term "dual tableaux". Dual tableaux are strongly motivated by R–S systems, but at the same time they are "genuine" tableaux, conceptualised as trees and in S-formulation. The term "dual" is to emphasize duality of R–S systems with respect to analytic tableau systems. The former may be interpreted as deriving Conjunctive Normal Form of a formula A, and is a "validity checker", whereas the later attempts to build a proof of A by deriving Disjunctive Normal Form of '$\neg A$' and is thus an "unsatisfiability checker" (the terminology comes from [30]).

1.5. Tableau System KE

In [16,17] the authors have presented a tableau system which analyses α- and β-formulas exclusively in a linear manner. For example, a β-formula of the form '$A \vee B$' is analysed only when one of: '$\neg A$' or '$\neg B$' is present at a branch. The presence of '$A \vee B$' and '$\neg A$' allows us to infer B, and similarly for the other case. Obviously, one needs a rule introducing the missing '$\neg A$'

or '$\neg B$' when necessary. And here comes the rule of cut—the only branching rule of the system.

The motivation for using cut in this way, in order to decrease the size of derivation trees by restricting the use of branching rules, goes back at least to [8]. If the size of a tree is supposed to model time complexity (see for example [31, p. 551] for this claim), then cut-based proof procedures seem to support efficiency.[3] As the authors state in [16,17], the rule of cut expresses the Principle of Bivalence, a principle which is absent in the formulation of standard analytic tableaux. The lack of this principle in the foundations of the method can lead, at least in some cases, to a computational collapse. The rule of cut is then introduced as a remedy.

1.6. Duality of Proof Procedures and at Least Two Shades of Cut

As we have said, this paper is a continuation of [14]. In [14] the authors have described the so-called erotetic calculi for the logics considered here (except for CPL with equivalence, which is new in this paper). Erotetic calculus is a calculus of questions originally developed in the framework of Inferential Erotetic Logic (see [37,38] and Section 6 of this paper). But the questions of a formal language are based on finite sequences of sequents and this is where the purely proof-theoretical perspective emerges. At the moment, we leave the erotetic aspect aside and analyse only sequents.

Basically, there are two types of erotetic calculi: canonical and dual. Canonical constructions are to a large degree erotetic versions of R–S systems. In [14] the authors have introduced calculi which are dual with respect to the canonical erotetic calculi. Duality may be expressed both in proof-theoretical and in semantic terms.

Proof-theoretically, the difference between the two types of calculi is in the nature of the closing conditions: these may be arrived at through a kind of decomposition of formulas and inspection of complementary formulas (canonical calculi), or through a decomposition and resolution (dual calculi). Semantically, the relation of duality can be expressed as follows. Suppose S is a finite sequence of formulas of a given formal language, and v is a valuation function defined for the language and with values in $\{0, 1\}$. Then we may define two semantical, dual to each other, properties: the first property consists in S having at least one term true under v, whereas the second one consists in S having at least one term false under v. The erotetic

[3] It seems, however, that there is no clear evidence that implementing cut leads to an improvement of a proof procedure in terms of efficiency. We are indebted to one of the Reviewers for spotting this problem.

rules of canonical calculi preserve the first property, and the dual erotetic calculi preserve the second property.

Let us emphasize that the duality relation between erotetic calculi is located on the level of proof-procedures, whereas duality of "dual tableaux" may be explained in terms of the two sides of a sequent: the analysis characteristic of analytic tableaux takes place on the left side of sequents, whereas the R–S derivation process, and the analysis specific to dual tableaux, takes place on the right side.

Therefore there are at least two relations of duality worth study. One is located at the level of formulas in the sense of the difference between αs and βs; in other words, it is the level of two sides of a sequent. The second relation is at the level of proof-procedures; it is what we do with the sequents and how we arrive at the conclusion that a proof has been obtained. Finally, it is worth to stress that on the second level of duality the cut rule known from sequent calculi is "canonical" and, as was said, it expresses the Principle of Bivalence, whereas the resolution rule is its dual and it expresses the Principle of Non-contradiction.

2. Rasiowa–Sikorski Deduction System for Classical Propositional Logic

The language $\mathcal{L}_{\mathsf{CPL}}$ of Classical Propositional Logic (CPL for short) consists of countably infinitely many propositional variables $p_1, p_2, \ldots, p_i, \ldots$, logical connectives $\neg, \wedge, \vee, \rightarrow$ and parentheses (,). We use VAR for the set of propositional variables and write p, q, r, s instead of p_1, p_2, p_3, p_4. The notion of formula of $\mathcal{L}_{\mathsf{CPL}}$ is defined in a standard way, $\mathrm{FOR}^{\mathsf{CPL}}$ stands for the set of all formulas of $\mathcal{L}_{\mathsf{CPL}}$. We will use A, B, C as metavariables for formulas of $\mathcal{L}_{\mathsf{CPL}}$. For simplicity, single quotation marks (*i.e.*: ' ') will be used in two roles: to indicate that an expression is mentioned (not used) and as Quinean corners. We also resign from the use of it whenever there is no risk of a misunderstanding.

Language $\mathcal{L}_{\mathsf{CPL}}$ is equipped with usual semantics based on Boolean valuations with 0 and 1 for false and truth, respectively. We shall write "valuation" instead of "Boolean valuation". Formulas true under every valuation are called CPL-valid.

In the sequel we will refer to the following fact, which we state without proof (see [19, p. 15]):

Table 1. α, β assignment for CPL

α	α_0	α_1	β	β_0	β_1
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B
$\neg(A \to B)$	A	$\neg B$	$A \to B$	$\neg A$	B

Table 2. κ assignment for CPL

κ	κ_0
$\neg\neg A$	A

Fact 1. *For any function f from the set* VAR *to the set* $\{0, 1\}$ *there exists exactly one valuation v which is an extension of f, that is, such that $v(p_i) = f(p_i)$ for each $p_i \in$ VAR.*

We use the uniform notation as introduced in [34]. The following table (see Table 1) defines the meaning of αs and βs for CPL.
We also decide to treat doubly negated formulas separately, and thus introduce the κ-assignment presented in Table 2.[4]

Quite obviously:

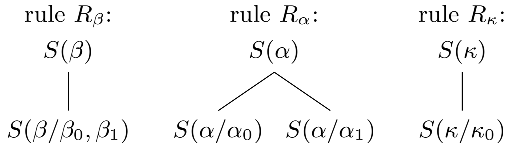
Corollary 1. *In the case of α-, β-, and κ-formulas defined for* CPL*, for an arbitrary valuation v,*

1. $v(\alpha) = 1$ *iff* $v(\alpha_0) = 1$ *and* $v(\alpha_1) = 1$*,*
2. $v(\beta) = 0$ *iff* $v(\beta_0) = 0$ *and* $v(\beta_1) = 0$*,*
3. $v(\kappa) = v(\kappa_0)$*.*

Letters S, T will refer to finite sequences of formulas of language $\mathcal{L}_{\mathsf{CPL}}$. We will use the sign: $'$ for concatenation of finite sequences, thus '$S\,'\,T$' refers to the result of concatenation of S and T. In the case of one-term sequences, we will often omit angle brackets. For example, we will write '$S\,'\,A\,'\,T$' instead of '$S\,'\,\langle A\rangle\,'\,T$'. Also the following convention will be useful. The inscription '$S(A)$' will refer to a finite sequence of formulas of $\mathcal{L}_{\mathsf{CPL}}$ such that A is its term (element). In other words, we can say that a sequence is of the form '$S(A)$', or that it is of the form '$S_1\,'\,A\,'\,S_2$', and in both cases we refer to the same class of sequences. By and large, the convention referring to

[4] The κ-notation has been introduced in [39] in order to account for a group of cases for quantifiers together with the case of double negation. The leading idea is that κs account for simple linear cases where one formula is transformed into one formula. This is how we use this notation in this paper.

Table 3. The rules of $\mathbb{RS}^{\mathsf{CPL}}$



concatenation is more precise but in some contexts, like proofs of theorems, less perspicuous than the one with parentheses, therefore we will use both conventions.

In order to indicate that both 'A' and 'B' are terms of S we shall write '$S(A)(B)$'; the order of A and B in S is arbitrary, thus '$S(A)(B)$' and '$S(B)(A)$' have the same meaning. Then by '$S(A/B)$' we mean the result of replacing the one distinguished term of S of the form 'A' with an occurrence of 'B'. In other words, if '$S(A)$' is of the form '$S_1 \,'\, A \,'\, S_2$' (in the concatenation-convention), then '$S(A/B)$' refers to '$S_1 \,'\, B \,'\, S_2$'. Observe that S_1 and S_2 may contain occurrences of A, and these occurrences are not replaced. The inscription '$S(A_1/B_1)(A_2/B_2)$' will refer to a superposition of two replacement operations, and so on. We will also need: '$S(A/B_1, B_2)$' which refers to the result of replacing the distinguished term of the form 'A' with two terms: B_1, B_2, that is, to '$S_1 \,'\, \langle B_1, B_2\rangle \,'\, S_2$'.

R–S deduction system for CPL will be called $\mathbb{RS}^{\mathsf{CPL}}$. It consists of the rules falling under the schemas displayed in Table 3.

If two formulas are of the forms: 'A' and '$\neg A$', then we call them *complementary*. A sequence of formulas containing a pair of complementary formulas among its terms will be called *fundamental*.

The tableaux built using R–S system tools will be called decomposition diagrams. Formally:

Definition 1. Let S be a finite sequence of formulas of language $\mathcal{L}_{\mathsf{CPL}}$. By a *decomposition diagram of S via the rules of* $\mathbb{RS}^{\mathsf{CPL}}$ we mean a finite tree labelled with finite sequences of formulas of $\mathcal{L}_{\mathsf{CPL}}$, where the labels are regulated by the rules of $\mathbb{RS}^{\mathsf{CPL}}$ and S labels the root.

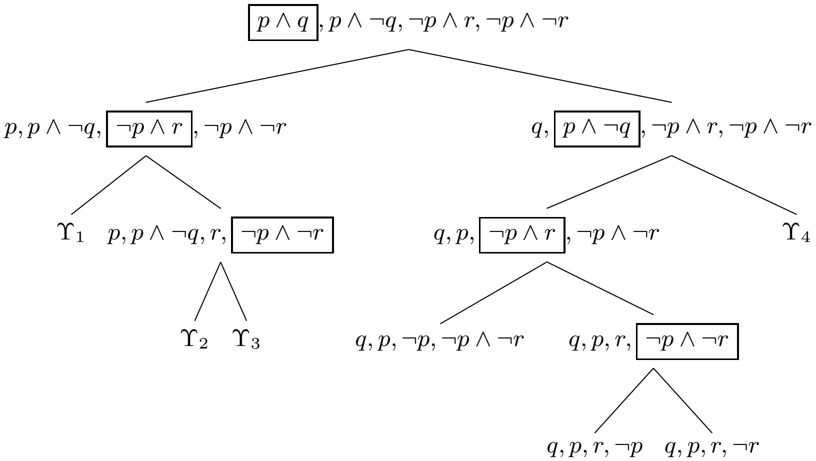
By a *proof of a formula, A, in* $\mathbb{RS}^{\mathsf{CPL}}$ we mean a decomposition diagram of the one-term sequence $\langle A\rangle$ via the rules of $\mathbb{RS}^{\mathsf{CPL}}$ each leaf of whose is labelled with a fundamental sequence.

The following example presents a decomposition diagram of the sequence: $\langle p \wedge q, p \wedge \neg q, \neg p \wedge r, \neg p \wedge \neg r\rangle$ via the rules of $\mathbb{RS}^{\mathsf{CPL}}$. In the sequel, we omit the angle brackets around sequences. The formulas which are acted upon by a rule are boxed. Every time R_α is applied.

Table 4. ε assignment for CPL

ε	ε_{00}	ε_{01}	ε_{10}	ε_{11}
$A \leftrightarrow B$	A	$\neg B$	$\neg A$	B
$\neg(A \leftrightarrow B)$	$\neg A$	$\neg B$	A	B

Example 1.


$\boxed{p \wedge q}, p \wedge \neg q, \neg p \wedge r, \neg p \wedge \neg r$
$p, p \wedge \neg q, \boxed{\neg p \wedge r}, \neg p \wedge \neg r$
$q, \boxed{p \wedge \neg q}, \neg p \wedge r, \neg p \wedge \neg r$
Υ_1
$p, p \wedge \neg q, r, \boxed{\neg p \wedge \neg r}$
$q, p, \boxed{\neg p \wedge r}, \neg p \wedge \neg r$
Υ_4
Υ_2
Υ_3
$q, p, \neg p, \neg p \wedge \neg r$
$q, p, r, \boxed{\neg p \wedge \neg r}$
$q, p, r, \neg p$
$q, p, r, \neg r$

where $\Upsilon_1 = p, p \wedge \neg q, \neg p, \neg p \wedge \neg r$, $\Upsilon_2 = p, p \wedge \neg q, r, \neg p$, $\Upsilon_3 = p, p \wedge \neg q, r, \neg r$, $\Upsilon_4 = q, \neg q, \neg p \wedge r, \neg p \wedge \neg r$.

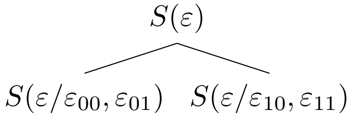
Assume that we add the equivalence connective '$\leftrightarrow$' to the language. The extended language will be called '$\mathcal{L}_{\mathsf{CPL}\leftrightarrow}$', and the Classical Propositional Logic expressed in $\mathcal{L}_{\mathsf{CPL}\leftrightarrow}$ will be called '$\mathsf{CPL}(\leftrightarrow)$'. Then the assignment given in Table 4 occurs useful.

Let us observe that the following is true:

Corollary 2. *In the case of ε-formulas defined for $\mathsf{CPL}(\leftrightarrow)$, for an arbitrary valuation v:*

4. $v(\varepsilon) = 1$ *iff* $(v(\varepsilon_{00}) = 1$ *or* $v(\varepsilon_{01}) = 1)$ *and* $(v(\varepsilon_{10}) = 1$ *or* $v(\varepsilon_{11}) = 1)$.

The R–S system for logic $\mathsf{CPL}(\leftrightarrow)$, called $\mathbb{RS}^{\mathsf{CPL}\leftrightarrow}$, consists of the rules falling under schemas R_α, R_β, R_κ, and R_ε:


$S(\varepsilon)$
$S(\varepsilon/\varepsilon_{00}, \varepsilon_{01})$
$S(\varepsilon/\varepsilon_{10}, \varepsilon_{11})$

In order to demonstrate soundness of R–S systems we interpret sequences of formulas semantically as disjunctions.

Definition 2. A finite sequence S of formulas of language $\mathcal{L}_{\mathsf{CPL}}$ (language $\mathcal{L}_{\mathsf{CPL}\leftrightarrow}$) is *correct under valuation* v iff v assigns value 1 to at least one term of S.

Corollary 3. *A fundamental sequence is correct under every valuation.*

Corollaries 1 and 2 may be easily used to prove that the rules: R_α, R_β, R_κ, R_ε are *correct* (or *sound*) in the following sense: they preserve correctness under a valuation of sequences of formulas from a premise to a conclusion(s). The corollaries, however, may as well be used to show that the rules preserve correctness under a valuation in the opposite direction: from a conclusion(s) to a premise, that is, that they are semantically invertible. To sum up:

Lemma 1. *Let S and T represent a premise and a conclusion (respectively) of a rule falling under schema R_β or R_κ. For any valuation v: S is correct under v iff T is correct under v.*

Lemma 2. *Let S, T_0 and T_1 represent a premise and conclusions (respectively) of a rule falling under schema R_α or R_ε. For any valuation v: S is correct under v iff both T_0 and T_1 are correct under v.*

By Lemmas 1, 2 and Corollary 3 we obtain:

Theorem 1. (soundness of $\mathbb{RS}^{\mathsf{CPL}}$ and $\mathbb{RS}^{\mathsf{CPL}\leftrightarrow}$) *If there exists a proof of A in $\mathbb{RS}^{\mathsf{CPL}}$ (in $\mathbb{RS}^{\mathsf{CPL}\leftrightarrow}$), then A is* CPL*-valid.*

We will show completeness of the method in Section 5.

3. Rasiowa–Sikorski Deduction Systems for CLuN, CLuNs and mbC

We will formalize the three non-classical logics by the schemas of rules presented for CPL and CPL($\leftrightarrow$). We adopt all the language and notational conventions introduced in the previous section. Until the end of this section let $\mathsf{L} \in \{\mathsf{CLuN}, \mathsf{CLuNs}, \mathsf{mbC}\}$. For each L we will alter the definition of αs, βs, and possibly κs and εs. Therefore the tables introducing the uniform notation will present the proper meaning of the rule schemas.

Table 5. The axiom system for CLuN

$A \to (B \to A)$	$(A \to B) \to (((A \to (B \to C)) \to (A \to C))$
$A \to (B \to (A \wedge B))$	$(A \wedge B) \to A$
$(A \wedge B) \to B$	$A \to (A \vee B)$
$B \to (A \vee B)$	$(A \to C) \to ((B \to C) \to (A \vee B \to C))$
$A \vee (A \to B)$	$A \vee \neg A$
$A \to (\neg A \to B)$	$A \vee \sim A$
(**MP**) If $\vdash_{\mathsf{CLuN}} A$ and $\vdash_{\mathsf{CLuN}} A \to B$, then $\vdash_{\mathsf{CLuN}} B$	

Table 6. The axioms for the CLuNs-negation

$\sim\sim A \to A$	$A \to \sim\sim A$
$\sim (A \to B) \to A \wedge \sim B$	$A \wedge \sim B \to \sim (A \to B)$
$\sim (A \wedge B) \to \sim A \vee \sim B$	$\sim A \vee \sim B \to \sim (A \wedge B)$
$\sim (A \vee B) \to \sim A \wedge \sim B$	$\sim A \wedge \sim B \to \sim (A \vee B)$

3.1. CLuN, CLuNs and mbC: Axiomatic Account

Logics CLuN and CLuNs are expressed in the same language $\mathcal{L}_{\mathsf{CLuN}}$, which is built upon language $\mathcal{L}_{\mathsf{CPL}}$ by adding the sign '$\sim$' for paraconsistent negation. The set $\mathrm{FOR}^{\mathsf{CLuN}}$ of formulas of $\mathcal{L}_{\mathsf{CLuN}}$ is defined by the following BNF-grammar:

$$A ::= p \mid \neg A \mid \sim A \mid A \wedge A \mid A \vee A \mid A \to A$$

Let us recall that we consider a $\neg$-extension of the original system CLuN.

The language $\mathcal{L}_{\mathsf{mbC}}$ of the logic mbC is the language of CLuN enriched with the symbol '$\circ$' (*consistency operator*). The set $\mathrm{FOR}^{\mathsf{mbC}}$ of formulas of $\mathcal{L}_{\mathsf{mbC}}$ is defined by the following BNF-grammar:

$$A ::= p \mid \neg A \mid \sim A \mid \ \circ A \mid A \wedge A \mid A \vee A \mid A \to A$$

Logic mbC is usually worded in a language without classical negation, but it is possible to define the constant *falsum* in it by putting:

$$\bot ::= \ \circ A \wedge (A \wedge \sim A)$$

Then the classical negation is defined by: $\neg A ::= A \to \bot$. However, as we consider mbC as an extension of CLuN we take '$\neg$' as primitive.

Table 5 presents the Hilbert-style deductive system for CLuN. Axiom 12 can be equivalently stated as follows: $(A \to \sim A) \to \sim A$. The axiomatic account of CLuNs is obtained by adding the axioms presented in Table 6 to the axiomatic basis of CLuN.

In order to obtain the axiomatic characterization of mbC we add the following axiom to the axioms of CLuN:

$$\text{Ax}(\circ) \qquad \circ A \rightarrow (A \rightarrow ({\sim} A \rightarrow B))$$

3.2. CLuN, CLuNs and mbC: Semantics

Semantics of the logics CLuN, CLuNs and mbC is sometimes based on the notion of semivaluation.[5] For our purposes it is more convenient to use the notions introduced below.

Let $\text{FOR}_{\sim}^{\mathsf{CLuN}} = \{{\sim} A : A \in \text{FOR}^{\mathsf{CLuN}}\}$.

Definition 3. (CLuN-*valuation*) A CLuN-*valuation* is a function $v : \text{FOR}^{\mathsf{CLuN}} \longrightarrow \{0, 1\}$ satisfying the following conditions:

($v1$) $v(A \wedge B) = 1$ iff $v(A) = v(B) = 1$,

($v2$) $v(A \vee B) = 0$ iff $v(A) = v(B) = 0$,

($v3$) $v(A \rightarrow B) = 0$ iff $v(A) = 1$ and $v(B) = 0$,

($v4$) $v(\neg A) = 1$ iff $v(A) = 0$,

($v5$) there exists an assignment function $\overline{v} : \text{FOR}_{\sim}^{\mathsf{CLuN}} \longrightarrow \{0, 1\}$ such that $v({\sim} A) = 1$ iff $v(A) = 0$ or $\overline{v}({\sim} A) = 1$.

The idea is that the assignment $\overline{v}$ directly assigns a logical value to paraconsistently negated formulas independently of the value assigned by v to the arguments of paraconsistent negation. For this reason '$\sim A$' can be true although A is also true.

The notion of CLuNs-valuation is akin to the above, but with the difference in the definition of the direct $\overline{v}$-assignment. This time the formulas whose values are assigned "directly" are only those formulas of the form '$\sim A$', where either A is a propositional variable or A has the form '$\neg B$' for some B. Let $\text{FOR}_{\sim}^{\mathsf{CLuNs}} = \{{\sim} A : A \in \text{VAR}\} \cup \{{\sim} \neg A : A \in \text{FOR}^{\mathsf{CLuN}}\}$.

Definition 4. (CLuNs-*valuation*) A CLuNs-*valuation* is a function $v : \text{FOR}^{\mathsf{CLuN}} \longrightarrow \{0, 1\}$ satisfying conditions ($v1$)–($v4$) from Definition 3 and the following:

[5] The notion of CLuN-valuation is the basis of the standard approach to the semantics of CLuN (see [4,5]). The concept of CLuN-semivaluation is not explicitly analysed in the literature on CLuN and CLuNs, but the idea of semivaluation is present in the studies on LFIs [10]. The notion of mbC-semivaluation is described in [10] under the name "bivaluation semantics for mbC".

($v5^*$) there exists an assignment function $\overline{v} : \mathrm{FOR}^{\mathsf{CLuNs}}_{\sim} \longrightarrow \{0,1\}$ such that $v(\sim A) = 1$ iff $v(A) = 0$ or $\overline{v}(\sim A) = 1$,

($v6$) $v(\sim (A \vee B)) = v(\sim A \wedge \sim B)$,

($v7$) $v(\sim (A \rightarrow B)) = v(A \wedge \sim B)$,

($v8$) $v(\sim (A \wedge B)) = v(\sim A \vee \sim B)$,

($v9$) $v(\sim\sim A) = v(A)$.

For the case of mbC, let $\mathrm{FOR}^{\mathsf{mbC}}_{\sim\circ} = \{\sim A : A \in \mathrm{FOR}^{\mathsf{mbC}}\} \cup \{\circ A : A \in \mathrm{FOR}^{\mathsf{mbC}}\}$.

Definition 5. (mbC-*valuation*) An mbC-*valuation* is a function $v : \mathrm{FOR}^{\mathsf{mbC}} \longrightarrow \{0,1\}$ satisfying conditions ($v1$)–($v4$) from Definition 3 and the following:

($v5^{**}$) there exists an assignment function $\overline{v} : \mathrm{FOR}^{\mathsf{mbC}}_{\sim\circ} \longrightarrow \{0,1\}$ such that

$v(\sim A) = 1$ iff $v(A) = 0$ or $\overline{v}(\sim A) = 1$

and

$v(\circ A) = 1$ iff ($v(A) = 0$ and $\overline{v}(\circ A) = 1$) or ($v(\sim A) = 0$ and $\overline{v}(\circ A) = 1$)

3.3. CLuN, CLuNs and mbC: Rules

Logics L will be formalized in a language built upon $\mathcal{L}_{\mathsf{L}}$ by adding the following operator: 'χ'. Called "skyhook" the operator is used to express the fact that some formulas of $\mathcal{L}_{\mathsf{L}}$ can get the direct assignment of a logical value. This idea has been used in [40], and then in [14]. Here we also take this approach.

3.3.1. CLuN The language resulting from $\mathcal{L}_{\mathsf{CLuN}}$ by the addition of 'χ' will be denoted by the symbol '$\mathcal{L}_{\mathsf{CLuN}^+}$'. In order to avoid introducing new metavariables, we define the syntax of "+-languages" not in BNF-format, but as in [40]. The set $\mathrm{FOR}^{\mathsf{CLuN}+}$, of formulas of $\mathcal{L}_{\mathsf{CLuN}^+}$, is defined as the smallest set such that: (i) each formula of $\mathcal{L}_{\mathsf{CLuN}}$ is a formula of $\mathcal{L}_{\mathsf{CLuN}^+}$; (ii) if '$\sim A$' is a formula of $\mathcal{L}_{\mathsf{CLuN}}$, then '$\chi \sim A$' and '$\neg\chi \sim A$' are formulas of $\mathcal{L}_{\mathsf{CLuN}^+}$.

The notion of CLuN-valuation for the richer language $\mathcal{L}_{\mathsf{CLuN}^+}$ is obtained from the definition of CLuN-valuation for the language $\mathcal{L}_{\mathsf{CLuN}}$ by the addition of the following two clauses:

(χ) $v(\chi \sim A) = 1$ iff $\overline{v}(\sim A) = 1$

($\neg\chi$) $v(\neg\chi \sim A) = 1$ iff $\overline{v}(\sim A) = 0$

Table 7. α, β assignment for CLuN

α	α_0	α_1	β	β_0	β_1
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B
$\neg(A \rightarrow B)$	A	$\neg B$	$A \rightarrow B$	$\neg A$	B
$\neg \sim A$	A	$\neg\chi \sim A$	$\sim A$	$\neg A$	$\chi \sim A$

Table 7 presents the α, β assignment for logic CLuN; κ is understood like in CPL (see Table 2), ε does not apply to CLuN.

As in the classical case, the following holds:

Lemma 3. *In the case of α-, β-, and κ-formulas defined for* CLuN*, for an arbitrary* CLuN*-valuation v of language $\mathcal{L}_{\mathsf{CLuN}^+}$,*

1. $v(\alpha) = 1$ *iff* $v(\alpha_0) = 1$ *and* $v(\alpha_1) = 1$*,*
2. $v(\beta) = 0$ *iff* $v(\beta_0) = 0$ *and* $v(\beta_1) = 0$*,*
3. $v(\kappa) = v(\kappa_0)$*.*

Proof. We consider only one case. Let $\alpha = \neg \sim A$. Let v stand for an arbitrary CLuN-valuation and suppose that $v(\alpha) = 1$. Then $v(\sim A) = 0$ by condition ($v4$) of Definition 3. By condition ($v5$), there exists an assignment $\overline{v}$ such that $v(\sim A) = 1$ iff $v(A) = 0$ or $\overline{v}(\sim A) = 1$. This entails that $v(A) = 1$ and, by ($\neg\chi$), $v(\neg\chi \sim A) = 1$. We have shown that if $v(\alpha) = 1$, then $v(\alpha_0) = 1$ and $v(\alpha_1) = 1$. ■

The R–S system formalizing logic CLuN, named $\mathbb{RS}^{\mathsf{CLuN}}$, is composed of rules falling under the schemas R_α, R_β and R_κ, with αs and βs defined by Table 7 and κs defined by Table 2.

3.3.2. CLuNs

In the case of logic CLuNs, we make use of the same language $\mathcal{L}_{\mathsf{CLuN}^+}$, but redefine the notion of formula and the uniform notation. Thus let $\mathcal{L}_{\mathsf{CLuNs}^+} = \mathcal{L}_{\mathsf{CLuN}^+}$. The set $\mathrm{FOR}^{\mathsf{CLuNs}+}$, of formulas of $\mathcal{L}_{\mathsf{CLuNs}^+}$, is defined as the smallest set such that: (i) each formula of $\mathcal{L}_{\mathsf{CLuNs}}$ is a formula of $\mathcal{L}_{\mathsf{CLuNs}^+}$; (ii) if '$\sim A$' is a formula of $\mathcal{L}_{\mathsf{CLuNs}}$ and either $A \in \mathrm{VAR}$ or A is of the form '$\neg B$', then '$\chi \sim A$' and '$\neg\chi \sim A$' are formulas of $\mathcal{L}_{\mathsf{CLuNs}^+}$.

Table 8 presents the α, β assignment for CLuNs. The assignments for '$\sim A$' and '$\neg \sim A$' are correct iff A is a propositional variable or a classically negated formula.

Table 9 presents the κ assignment for CLuNs. Again, ε does not apply here.

Table 8. α, β assignment for CLuNs

α	α_0	α_1	β	β_0	β_1
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B
$\neg(A \rightarrow B)$	A	$\neg B$	$A \rightarrow B$	$\neg A$	B
$\sim (A \vee B)$	$\sim A$	$\sim B$	$\neg \sim (A \vee B)$	$\neg \sim A$	$\neg \sim B$
$\sim (A \rightarrow B)$	A	$\sim B$	$\neg \sim (A \rightarrow B)$	$\neg A$	$\neg \sim B$
$\neg \sim (A \wedge B)$	$\neg \sim A$	$\neg \sim B$	$\sim (A \wedge B)$	$\sim A$	$\sim B$
for $A \in$ VAR or $A = \neg B$ $\neg \sim A$	A	$\neg\chi \sim A$	for $A \in$ VAR or $A = \neg B$ $\sim A$	$\neg A$	$\chi \sim A$

Table 9. κ assignment for CLuNs

κ	κ_0
$\neg\neg A$	A
$\sim\sim A$	A
$\neg\sim\sim A$	$\neg A$

The notion of CLuNs-valuation is extended to language $\mathcal{L}_{\mathsf{CLuNs}^+}$ by adding conditions (χ), $(\neg\chi)$, just as in the case of CLuN, with the exception that 'χA' and '$\neg\chi A$' are formulas of $\mathcal{L}_{\mathsf{CLuNs}^+}$.

We state the following without proof:

Lemma 4. *In the case of α-, β- and κ-formulas defined for* CLuNs*, for an arbitrary* CLuNs*-valuation v of language $\mathcal{L}_{\mathsf{CLuNs}^+}$,*

1. $v(\alpha) = 1$ *iff* $v(\alpha_0) = 1$ *and* $v(\alpha_1) = 1$*,*
2. $v(\beta) = 0$ *iff* $v(\beta_0) = 0$ *and* $v(\beta_1) = 0$*,*
3. $v(\kappa) = v(\kappa_0)$*.*

The R–S system formalizing logic CLuNs, called $\mathbb{RS}^{\mathsf{CLuNs}}$, is composed of the rules falling under the schemas R_α, R_β, R_κ with αs and βs defined by Table 8 and κs defined by Table 9.

3.3.3. mbC Let $\mathcal{L}_{\mathsf{mbC}^+}$ be the language obtained from the language $\mathcal{L}_{\mathsf{mbC}}$ by the addition of 'χ'. The set $\mathrm{FOR}^{\mathsf{mbC}+}$, of formulas of this language, is the smallest set such that: (i) each formula of $\mathcal{L}_{\mathsf{mbC}}$ is a formula of $\mathcal{L}_{\mathsf{mbC}^+}$; (ii) if A is a formula of $\mathcal{L}_{\mathsf{mbC}}$, where $A = {\sim} B$ or $A = \circ B$, then 'χA' and '$\neg\chi A$' are formulas of $\mathcal{L}_{\mathsf{mbC}^+}$.

Since logic mbC is built upon CLuN, the notions of α- and β-formulas are defined by Table 7. In addition, we redefine the notion of ε-formulas.

Table 10. ε assignment for mbC

ε	ε_{00}	ε_{01}	ε_{10}	ε_{11}
$\circ A$	$\neg A$	$\neg \sim A$	$\chi \circ A$	$\chi \circ A$
$\neg \circ A$	A	$\neg\chi \circ A$	$\neg\chi \circ A$	$\sim A$

The notion of mbC-valuation for the language $\mathcal{L}_{\mathsf{mbC}^+}$ is obtained from the definition of mbC-valuation for the language $\mathcal{L}_{\mathsf{mbC}}$ by the addition of the following four clauses:

(χ) $v(\chi \sim A) = 1$ iff $\overline{v}(\sim A) = 1$

$(\neg\chi)$ $v(\neg\chi \sim A) = 1$ iff $\overline{v}(\sim A) = 0$

(χ') $v(\chi\circ A) = 1$ iff $\overline{v}(\circ A) = 1$

$(\neg\chi')$ $v(\neg\chi\circ A) = 1$ iff $\overline{v}(\circ A) = 0$

Again, we have what follows:

Lemma 5. *In the case of α-, β-, κ- and ε-formulas defined for* mbC*, for an arbitrary* mbC*-valuation v of language $\mathcal{L}_{\mathsf{mbC}^+}$,*

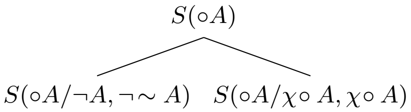
1. $v(\alpha) = 1$ *iff* $v(\alpha_0) = 1$ *and* $v(\alpha_1) = 1$*,*
2. $v(\beta) = 0$ *iff* $v(\beta_0) = 0$ *and* $v(\beta_1) = 0$*,*
3. $v(\kappa) = v(\kappa_0)$*,*
4. $v(\varepsilon) = 1$ *iff* $(v(\varepsilon_{00}) = 1$ *or* $v(\varepsilon_{01}) = 1)$ *and* $(v(\varepsilon_{10}) = 1$ *or* $v(\varepsilon_{11}) = 1)$*.*

Proof. This time we consider only the $(\Leftarrow)$ direction for $\varepsilon = \circ A$. Suppose that $v(\varepsilon_{00}) = v(\neg A) = 1$ and $v(\varepsilon_{10}) = v(\chi \circ A) = 1$. Then by clause (χ'), clause $(v4)$ of Definition 3 and clause $(v5^{**})$ of Definition 5, also $v(\circ A) = 1$. The case of $v(\varepsilon_{00}) = v(\varepsilon_{11}) = 1$ is exactly the same. Suppose that $v(\varepsilon_{01}) = v(\neg \sim A) = 1$ and $v(\varepsilon_{10}) = v(\chi \circ A) = 1$. Then, again, by clause (χ'), clause $(v4)$ of Definition 3 and clause $(v5^{**})$ of Definition 5, $v(\circ A) = 1$. The case $v(\varepsilon_{01}) = v(\varepsilon_{11}) = 1$ is the same. ■

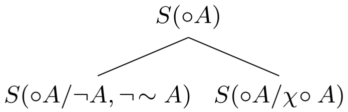
The R–S system for logic mbC, called $\mathbb{RS}^{\mathsf{mbC}}$, has all the rules of $\mathbb{RS}^{\mathsf{CLuN}}$ and the rules of the form R_ε. αs, βs and κs are defined as in the case of CLuN, ε is defined by Table 10.

There is a certain price we pay for the general description of $\mathsf{CPL}(\leftrightarrow)$ and mbC by the use of ε assignment. Namely, for $\varepsilon = \circ A$ we have a repetition in the right conclusion, as the rule R_ε takes the form depicted on the left below. It may be shown, however, that the rule $R^*_\circ$ depicted on the right is admissible in $\mathbb{RS}^{\mathsf{mbC}}$. We go back to this issue at the end of this section.

rule $R_\circ$:


$S(\circ A)$
$S(\circ A/\neg A, \neg\sim A)$
$S(\circ A/\chi\circ A, \chi\circ A)$

rule $R^*_\circ$:



The notions of decomposition diagram and proof are defined analogously as in the classical case. Observe, however, that the notion of proof is reserved for formulas of "pure" language $\mathcal{L}_{\mathsf{L}}$. The skyhook connective 'χ' may be introduced in the course of decomposition only.

Definition 6. Let S be a finite sequence of formulas of language $\mathcal{L}_{\mathsf{L}^+}$. By a *decomposition diagram of S via the rules of* $\mathbb{RS}^{\mathsf{L}}$ we mean a finite tree labelled with finite sequences of formulas of $\mathcal{L}_{\mathsf{L}^+}$, where the labels are regulated by the rules of $\mathbb{RS}^{\mathsf{L}}$ and S labels the root.

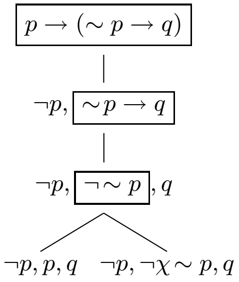
Let A be a formula of language $\mathcal{L}_{\mathsf{L}}$. By a *proof of a formula, A, in* $\mathbb{RS}^{\mathsf{L}}$ we mean a decomposition diagram of the one-term sequence $\langle A\rangle$ via the rules of $\mathbb{RS}^{\mathsf{L}}$ each leaf of whose is labelled with a fundamental sequence. If there exists a proof of formula A in $\mathbb{RS}^{\mathsf{L}}$, then we say that A *is provable in* $\mathbb{RS}^{\mathsf{L}}$.

The following generalization of the notion of proof will be used in the sequel.

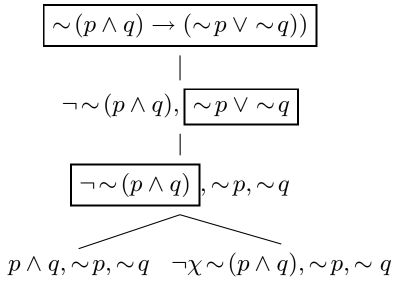
Definition 7. Let S be a finite sequence of formulas of language $\mathcal{L}_{\mathsf{L}^+}$ and let $\mathcal{T}$ be a decomposition diagram of S via the rules of $\mathbb{RS}^{\mathsf{L}}$. We say that $\mathcal{T}$ is *successful* iff each leaf of $\mathcal{T}$ is labelled with a fundamental sequence.

Here are some examples.

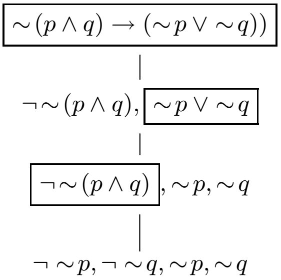
Example 2. Formula $p \rightarrow (\sim p \rightarrow q)$ is not provable in any of the systems presented here.


$\boxed{p \rightarrow (\sim p \rightarrow q)}$
$\neg p, \boxed{\sim p \rightarrow q}$
$\neg p, \boxed{\neg\sim p}, q$
$\neg p, p, q$
$\neg p, \neg\chi\sim p, q$

Example 3. Formula $\sim (p \wedge q) \rightarrow (\sim p \vee \sim q)$ is not provable in $\mathbb{RS}^{\mathsf{CLuN}}$, as there are no means to decompose formula '$\neg\chi \sim (p \wedge q)$'.


$\boxed{\sim (p \wedge q) \rightarrow (\sim p \vee \sim q))}$
$\neg \sim (p \wedge q), \boxed{\sim p \vee \sim q}$
$\boxed{\neg \sim (p \wedge q)}, \sim p, \sim q$
$p \wedge q, \sim p, \sim q$
$\neg\chi \sim (p \wedge q), \sim p, \sim q$

However, the same formula is provable in $\mathbb{RS}^{\mathsf{CLuNs}}$.


$\boxed{\sim (p \wedge q) \rightarrow (\sim p \vee \sim q))}$
$\neg \sim (p \wedge q), \boxed{\sim p \vee \sim q}$
$\boxed{\neg \sim (p \wedge q)}, \sim p, \sim q$
$\neg \sim p, \neg \sim q, \sim p, \sim q$

The following holds:

Corollary 4. *A fundamental sequence is correct under every* L*-valuation.*

Soundness of $\mathbb{RS}^{\mathsf{L}}$ is proved as in the classical case, by using Lemmas 3, 4, 5 and Corollary 4.

Theorem 2. (soundness of $\mathbb{RS}^{\mathsf{L}}$) *Let A be a formula of $\mathcal{L}_{\mathsf{L}}$. If there exists a proof of A in $\mathbb{RS}^{\mathsf{L}}$, then A is* L*-valid.*

At the end of this section, let us go back to the notion of admissibility of a rule in $\mathbb{RS}^{\mathsf{L}}$. By '$\mathbb{RS}^{\mathsf{L}} + R$' we mean the set of rules of $\mathbb{RS}^{\mathsf{L}}$ enlarged with the rules falling under the schema R. The notion of admissibility will be understood as follows.

Definition 8. We say that a rule R is *admissible in R–S system* $\mathbb{RS}^{\mathsf{L}}$ iff for each finite sequence S of formulas of $\mathcal{L}_{\mathsf{L}+}$, whenever there exists a successful decomposition diagram of S in $\mathbb{RS}^{\mathsf{L}} + R$, then there exists also a successful decomposition diagram of S in $\mathbb{RS}^{\mathsf{L}}$.

Generally, in order to prove admissibility of a rule in a deduction system it is enough to show that, first, the rule cannot serve to prove something that should not be provable, and second, the deduction system without the rule is complete. In the R–S setting the first property amounts to invertibility of the rule. It is easily seen that rule $R^*_\circ$ is invertible for the same reason $R_\circ$ is. Hence, after we prove completeness of $\mathbb{RS}^{\mathsf{L}}$ it will be easily seen that $R^*_\circ$ is admissible in $\mathbb{RS}^{\mathsf{L}}$. Also the following rule of cut (which will be called R_{cut}):

$$\begin{array}{c} S'\,T \\ \swarrow \qquad \searrow \\ S'\,A'\,T \quad S'\,\neg A'\,T \end{array}$$

may be shown to be admissible in $\mathbb{RS}^{\mathsf{L}}$ by using the same argument.

It is commonly believed that direct, constructive proofs of admissibility are more valuable than indirect ones. In our case, however, the direct proof amounts to delivering a procedure of proof-search in $\mathbb{RS}^{\mathsf{L}}$, but this seems simply trivial; the rules are sound, invertible and clearly they reduce complexity of formulas, thus almost any algorithm of the rules application will do. To sum up, we stay with the observation that both an indirect and a direct proof of admissibility of rules $R^*_\circ$, R_{cut} in $\mathbb{RS}^{\mathsf{L}}$ are obtainable.

4. The Rule of Cut

In this section we introduce the promised second variant of R–S systems. First, we introduce the following conventions. Until the end of this section let $\mathsf{L} \in \{\mathsf{CPL}, \mathsf{CPL}(\leftrightarrow), \mathsf{CLuN}, \mathsf{CLuNs}, \mathsf{mbC}\}$. In the case of $\mathsf{L} \in \{\mathsf{CPL}, \mathsf{CPL}(\leftrightarrow)\}$, the language $\mathcal{L}_{\mathsf{L}^+}$ equals $\mathcal{L}_{\mathsf{CPL}}/\mathcal{L}_{\mathsf{CPL}\leftrightarrow}$. If $A \in \mathcal{L}_{\mathsf{L}^+}$, then by $\overline{A}$ we refer to *complement of* A, that is

$$\overline{A} = \begin{cases} B, & \text{if} \quad A \text{ is of the form } \neg B \\ \neg A, & \text{otherwise} \end{cases}$$

Propositional variables and their negations are called *literals*.

The rules of $\mathbb{RS}^{\mathsf{CPL}}_{cut}$ follow under the schemas presented in Table 11. The letter 'l' in the names of the rules lR_α, lR_ε is for "*linear*".

Observe that rule lR_α may be applied provided there is an occurrence of '$\overline{\alpha_i}$' in the sequence. In the conclusion: '$S(\alpha/\alpha_j)$' only α has been replaced, the term of the form '$\overline{\alpha_i}$' does not disappear from S. A similar observation pertains to the rule for ε-formulas: there must be a term of the form '$\overline{\varepsilon_{ik}}$' in the premise, and in '$S(\varepsilon/\varepsilon_{jn}, \varepsilon_{jm})$' only ε is replaced.

Table 11. rules of $\mathbb{RS}^{\mathsf{CPL}}_{cut}$

rule lR_α: $\begin{array}{c} S(\alpha)(\overline{\alpha_i}) \\ \mid \\ S(\alpha/\alpha_j) \end{array}$ where $i, j \in \{0, 1\}$ $i \neq j$	rule R_β: $\begin{array}{c} S(\beta) \\ \mid \\ S(\beta/\beta_0, \beta_1) \end{array}$	rule lR_ε: $\begin{array}{c} S(\varepsilon)(\overline{\varepsilon_{ik}}) \\ \mid \\ S(\varepsilon/\varepsilon_{jn}, \varepsilon_{jm}) \end{array}$ where $i, j, k, n, m \in \{0, 1\}$ $i \neq j$ and $n \neq m$
rule R_κ: $\begin{array}{c} S(\kappa) \\ \mid \\ S(\kappa/\kappa_0) \end{array}$	the rule of cut R_{cut}: $\begin{array}{c} S'\,T \\ \swarrow \quad \searrow \\ S'\,A'\,T \quad S'\,\neg A'\,T \end{array}$	

Here is an example: a decomposition diagram of sequence $p \wedge q, p \wedge \neg q, \neg p \wedge r, \neg p \wedge \neg r$ via the rules of $\mathbb{RS}^{\mathsf{CPL}}_{cut}$ (see Example 4). The first rule applied is that of cut, therefore there is no formula "acted upon" in the first sequence.

Example 4.

$$\begin{array}{cc}
\multicolumn{2}{c}{p \wedge q, p \wedge \neg q, \neg p \wedge r, \neg p \wedge \neg r} \\
\swarrow & \searrow \\
p \wedge q, p \wedge \neg q, \boxed{\neg p \wedge r}, \neg p \wedge \neg r, \boxed{p} & \boxed{p \wedge q}, p \wedge \neg q, \neg p \wedge r, \neg p \wedge \neg r, \boxed{\neg p} \\
\mid & \mid \\
p \wedge q, p \wedge \neg q, r, \boxed{\neg p \wedge \neg r}, \boxed{p} & q, \boxed{p \wedge \neg q}, \neg p \wedge r, \neg p \wedge \neg r, \boxed{\neg p} \\
\mid & \mid \\
p \wedge q, p \wedge \neg q, r, \neg r, p & q, \neg q, \neg p \wedge r, \neg p \wedge \neg r, \neg p
\end{array}$$

Let us note that there is another possibility to obtain a fundamental sequence on the left branch of the above diagram. Not only '$\neg p \wedge \neg r$' and 'p' match the scheme of lR_α, but also '$\neg p \wedge \neg r$' and 'r' do. Here is an alternative for the last two nodes on the left branch of the diagram above:

$$\begin{array}{c}
p \wedge q, p \wedge \neg q, \boxed{r}, \boxed{\neg p \wedge \neg r}, p \\
\mid \\
p \wedge q, p \wedge \neg q, r, \neg p, p
\end{array}$$

Naturally, proof-search in $\mathbb{RS}^{\mathsf{CPL}}_{cut}$ is goal-directed and non-deterministic. The question about a general strategy for constructing minimal successful decomposition diagrams is very important, however, even an attempt to answer it goes beyond the scope of this paper.

For logic $\mathsf{CPL}(\leftrightarrow)$ we need linear rules for ε-formulas. The solution is lR_ε (see Table 11). As the Reader may expect, our aim is to produce the same schema for '$\leftrightarrow$' and '$\circ$'. The above schema of lR_ε for $\mathsf{CPL}(\leftrightarrow)$ produces some repetitions, however. *E.g.*, for $\varepsilon = A \leftrightarrow B$ and $i = 0, k = 0, j = 1, n = 0, m = 1$:

$$\begin{array}{c} S(A \leftrightarrow B)(\neg A) \\ | \\ S(A \leftrightarrow B/\neg A, B) \end{array}$$

which means that there are at least two occurrences of '$\neg A$' in the conclusion. However, as in the case of $R^*_\circ$, it is easy to see that the variant of lR_ε without the repetition is admissible in $\mathbb{RS}^{\mathsf{CPL}}_{cut}$.

The other R–S systems with cut are obtained as follows.

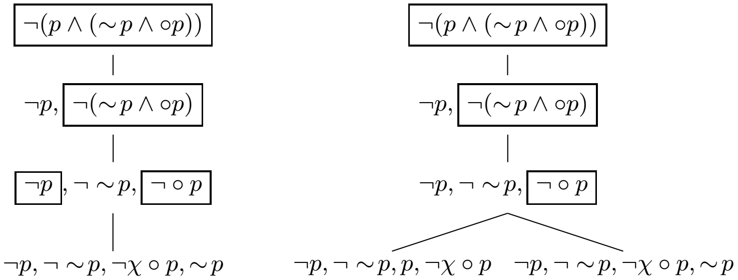
- $\mathbb{RS}^{\mathsf{CLuN}}_{cut}$: the rules lR_α, R_β, R_κ from Table 11 with α-, β-formulas defined as in Table 7 and κ-formulas defined as in Table 2.
- $\mathbb{RS}^{\mathsf{CLuNs}}_{cut}$: the rules lR_α, R_β, R_κ from Table 11 with α-, β-formulas defined as in Table 8, and κ-formulas defined as in Table 9.
- $\mathbb{RS}^{\mathsf{mbC}}_{cut}$: the rules lR_α, R_β, R_κ from Table 11 and rule lR_ε with α-, β-formulas defined as in Table 7, κ-formulas defined as in Table 2, and ε-formulas defined as in Table 10.

Here are some examples.

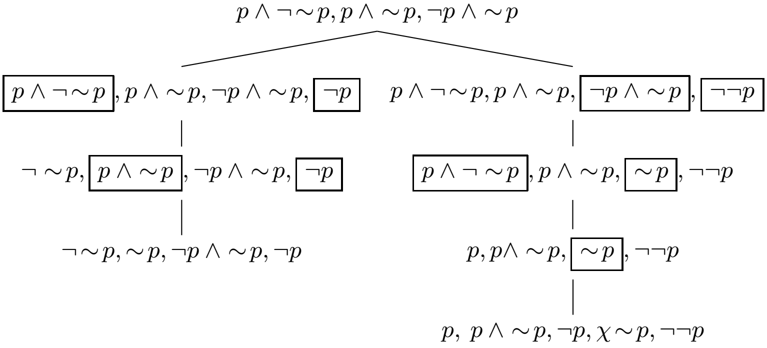
Example 5. The following is a proof of formula '$p \vee \sim p$' in $\mathbb{RS}^{\mathsf{CLuN}}_{cut}$. It is also a proof of the formula in $\mathbb{RS}^{\mathsf{CLuN}}$ since the only rule used is R_β which is common to both calculi.

$$\begin{array}{c} \boxed{p \vee \sim p} \\ | \\ p, \boxed{\sim p} \\ | \\ p, \neg p, \chi \sim p \end{array}$$

Example 6. Here is a proof of formula '$\neg(p \wedge (\sim p \wedge \circ p))$' in $\mathbb{RS}^{\mathsf{mbC}}_{cut}$ (to the left) and in $\mathbb{RS}^{\mathsf{mbC}}$ (to the right).



Example 7. A successful decomposition diagram of sequence $p \wedge \neg \sim p, p \wedge \sim p, \neg p \wedge \sim p$ via the rules of $\mathbb{RS}^{\mathsf{mbC}}_{cut}$. The Reader may check that the smallest successful decomposition diagram for this sequence via the rules of $\mathbb{RS}^{\mathsf{mbC}}$ has 5 branches.


p ∧ ¬∼p, p ∧ ∼p, ¬p ∧ ∼p
p ∧ ¬∼p, p ∧ ∼p, ¬p ∧ ∼p, ¬p
¬ ∼p, p ∧ ∼p, ¬p ∧ ∼p, ¬p
¬∼p, ∼p, ¬p ∧ ∼p, ¬p
p ∧ ¬∼p, p ∧ ∼p, ¬p ∧ ∼p, ¬¬p
p ∧ ¬ ∼p, p ∧ ∼p, ∼p, ¬¬p
p, p∧ ∼p, ∼p, ¬¬p
p, p ∧ ∼p, ¬p, χ∼p, ¬¬p

4.1. Analytic Restriction

A natural question to ask is that about analytic restriction of $\mathbb{RS}^{\mathsf{L}}_{cut}$. We introduce a version of the notion of analyticity which is, in a way, adjusted to formalism presented in this work. First, instead of the traditional notion of subformula we shall use the notion of decomposition set of a formula.

Definition 9. (*Decomposition set*) Let A be a formula of $\mathcal{L}_{\mathsf{L}^+}$. By *decomposition set of* A, symbolically $Dec(A)$, we mean the smallest set satisfying the following conditions:

1. $A \in Dec(A)$,
2. if $\kappa \in Dec(A)$ then $\kappa_0 \in Dec(A)$,
3. if $\beta \in Dec(A)$, then $\beta_i \in Dec(A)$ for both $i = 0, 1$,
4. if $\alpha \in Dec(A)$, then $\alpha_i \in Dec(A)$ for both $i = 0, 1$,
5. for $\mathsf{L} \in \{\mathsf{CPL}(\leftrightarrow), \mathsf{mbC}\}$, if $\varepsilon \in Dec(A)$, then $\varepsilon_{ij} \in Dec(A)$ for each $i, j \in \{0, 1\}$.

Moreover, let $S = \langle A_1, \dots, A_n \rangle$ be a finite sequence of formulas of $\mathcal{L}_{\mathsf{L}^+}$. We set:

$$Dec(S) = Dec(A_1) \cup \dots \cup Dec(A_n)$$

Definition 10. (*analytic application of a rule, analytic restriction of* $\mathbb{RS}^{\mathsf{L}}_{cut}$) Let R stand for a rule of $\mathbb{RS}^{\mathsf{L}}_{cut}$. We say that *rule R has been applied analytically to sequence S* iff

- R is one of R_β, R_κ, lR_α, lR_ε and this application of R yields a sequence T such that each term of T belongs to $Dec(S)$,
- R is R_{cut} and this application of R yields $T\,'\,A\,'\,U$ and $T\,'\,\neg A\,'\,U$ such that $\{A, \neg A\} \cap Dec(T\ '\ U) \neq \emptyset$.

Moreover, by *analytic restriction of* $\mathbb{RS}^{\mathsf{L}}_{cut}$ we mean the set of rules falling under the schemas R_β, R_κ, lR_α, lR_ε, R_{cut} but restricted to their analytic applications.

To state the obvious, every application of R_β, R_κ, lR_α, lR_ε is analytic. Soundness of the analytic restriction of $\mathbb{RS}^{\mathsf{L}}_{cut}$ follows from soundness of $\mathbb{RS}^{\mathsf{L}}_{cut}$. Completeness of the analytic restriction will be considered in the next section.

5. Completeness

The proof of completeness theorem presented below is inspired by a construction introduced by Raymond Smullyan, and then developed by Melvin Fitting and used successfully in completeness proofs for the classical and many non-classical logics (see, *e.g.*, [18,19,34]). In this abstract approach families of sets called "consistency properties" are defined syntactically in a way which encodes semantic property of consistency. Showing that the encoding is correct is actually the main work to be done in order to flip the bridge between syntax and semantics.

The idea to use a "dual" construction, where *refutability properties* are introduced instead of *consistency properties*, has been developed successfully in doctoral dissertation by Szymon Chlebowski (see [12]) in order to prove completeness of erotetic calculi for the First-Order Logic.[6] However, the construction presented here is adjusted (mainly weakened) to the purpose

[6] The notion of *dual Hintikka set* has been used by Smullyan in [35]. In the erotetic context it has been used for the first time in [13].

of describing propositional logics; also the first author has made it more "sensitive" to the sequence-format characteristic to R–S systems.

Again, until the end of this section let $\mathsf{L} \in \{\mathsf{CPL}, \mathsf{CPL}(\leftrightarrow), \mathsf{CLuN}, \mathsf{CLuNs}, \mathsf{mbC}\}$. As before, in the case of $\mathsf{L} \in \{\mathsf{CPL}, \mathsf{CPL}(\leftrightarrow)\}$, $\mathcal{L}_{\mathsf{L}+} = \mathcal{L}_{\mathsf{CPL}}/\mathcal{L}_{\mathsf{CPL}\leftrightarrow}$.

Definition 11. (*Refutability property*) Let $\mathcal{F}$ be a family of finite sequences of formulas of $\mathcal{L}_{\mathsf{L}+}$, the empty sequence included. We say that $\mathcal{F}$ is a *refutability property for* L iff the following conditions are satisfied:

1. No $S \in \mathcal{F}$ is a fundamental sequence.
2. If $S(\alpha) \in \mathcal{F}$, then $S(\alpha/\alpha_i) \in \mathcal{F}$ for $i = 0$ or for $i = 1$.
3. If $S(\beta) \in \mathcal{F}$, then $S(\beta/\beta_0, \beta_1) \in \mathcal{F}$.
4. If $S(\kappa) \in \mathcal{F}$, then $S(\kappa/\kappa_0) \in \mathcal{F}$.
5. If $S(\varepsilon) \in \mathcal{F}$, then $S(\varepsilon/\varepsilon_{i0}, \varepsilon_{i1}) \in \mathcal{F}$ for $i = 0$ or for $i = 1$.

Example 8. The following is an example of refutability property for CPL:

$$\mathcal{F} = \{\langle p \vee q, p \wedge q, r, s\rangle, \langle p \vee q, p, r, s\rangle, \langle p, q, p \wedge q, r, s\rangle, \langle p, q, p, r, s\rangle\}$$

The technical notion of rank of a formula (and a sequence) will be used in the completeness proof. In the case of $\mathsf{L} \in \{\mathsf{CPL}, \mathsf{CPL}(\leftrightarrow)\}$, $\mathrm{FOR}^{\mathsf{L}+} = \mathrm{FOR}^{\mathsf{CPL}}/\mathrm{FOR}^{\mathsf{CPL}\leftrightarrow}$.

Definition 12. (*Rank of a formula, rank of a sequence*) *Rank of a formula in* L, symbolically r_{L}, is a function $r_{\mathsf{L}} : \mathrm{FOR}^{\mathsf{L}+} \longrightarrow \mathbb{N}_0$ defined inductively as follows:

- $r_{\mathsf{L}}(p_i) = r_{\mathsf{L}}(\neg p_i) = 0$,
- if it applies, $r_{\mathsf{L}}(\chi A) = r_{\mathsf{L}}(\neg\chi A) = 0$,
- $r_{\mathsf{L}}(\kappa) = r_{\mathsf{L}}(\kappa_0) + 1$,
- $r_{\mathsf{L}}(\alpha) = r_{\mathsf{L}}(\alpha_0) + r_{\mathsf{L}}(\alpha_1) + 1$,
- $r_{\mathsf{L}}(\beta) = r_{\mathsf{L}}(\beta_0) + r_{\mathsf{L}}(\beta_1) + 1$,
- if it applies, $r_{\mathsf{L}}(\varepsilon) = \left(\sum\limits_{i,j \in \{0,1\}} r_{\mathsf{L}}(\varepsilon_{ij}) \right) + 1$.

If S is a finite, non-empty sequence of formulas of $\mathcal{L}_{\mathsf{L}+}$, then by *rank of* S, symbolically $r_{\mathsf{L}}(S)$, we mean:

$$r_{\mathsf{L}}(S) = \max\{r_{\mathsf{L}}(F) : F \text{ is a term of } S\}$$

It should be clear that the value of r_{L} depends on L, at least for some of the arguments. However, in order to simplify notation, we will write 'r' instead of 'r_{L}', as it should not cause any confusion.

In the case of $\mathsf{L} = \mathsf{CPL}$, by "$\mathsf{L}$-valuation" we mean Boolean valuation.

Definition 13. (*falsifying valuation*) Let S be a finite sequence of formulas of $\mathcal{L}_{\mathsf{L}+}$. If there is an L-valuation v such that each term of S is false under v, then we say that *v is a falsifying L-valuation of S* or that *S has a falsifying L-valuation.*

Let $\mathrm{FOR}^{\mathsf{L}}_{\chi}$ stand for the set of formulas of the form 'χA' which are formulas of language $\mathcal{L}_{\mathsf{L}+}$. Observe that the elements of the set $\mathrm{VAR} \cup \mathrm{FOR}^{\mathsf{L}}_{\chi}$ are semantically atomic (in semantics of L) and are pairwise logically independent (in semantics of L). This yields the following fact, which, similarly to the classical case, we state without proof:

Fact 2. *For any function f from the set $\mathrm{VAR} \cup \mathrm{FOR}^{\mathsf{L}}_{\chi}$ to the set $\{0, 1\}$ there exists exactly one L-valuation v which is an extension of f, that is, such that $v(A) = f(A)$ for each $A \in \mathrm{VAR} \cup \mathrm{FOR}^{\mathsf{L}}_{\chi}$.*

Lemma 6. (Counter-model existence lemma) *If a sequence, S, belongs to a refutability property for L, then S has a falsifying L-valuation.*

Proof. Let S be an arbitrary sequence which is an element of a refutability property $\mathcal{F}$ for logic L. If S is empty, then, trivially, each L-valuation is a falsifying L-valuation of S. For non-empty sequences the proof is by induction on rank of sequence S.

Base step: suppose that $r(S) = 0$, that is, S is a finite sequence of literals and/or formulas of the form 'χA', '$\neg\chi A$'. By clause 1. of Definition 11, there is no pair of complementary formulas among the terms of S. By Fact 2, the following assignment f of truth values:

$$\text{for each } A \in \mathrm{VAR} \cup \mathrm{FOR}^{\mathsf{L}}_{\chi}, f(A) = 0 \text{ iff } A \text{ is a term of } S$$

extends to an L-valuation on $\mathrm{FOR}^{\mathsf{L}+}$. Obviously, it is a falsifying L-valuation of S.

Induction hypothesis: each sequence from $\mathcal{F}$ of rank less than n has a falsifying L-valuation. Let $S \in \mathcal{F}$ and $r(S) = n$. There is at least one formula F in S of rank n. Suppose that there is exactly one such formula.

If F is an α-formula, then S is of the form '$S(\alpha)$'. By item 2. of Definition 11, '$S(\alpha/\alpha_0)$' $\in \mathcal{F}$ or '$S(\alpha/\alpha_1)$' $\in \mathcal{F}$, where α_0 and α_1 are the components of F. By definition of rank of a formula, $r(\alpha_i) < r(F)$ for $i \in \{0, 1\}$. In both cases, $r(S(\alpha/\alpha_i)) < n$, thus by induction hypothesis there exists a falsifying L-valuation v for '$S(\alpha/\alpha_i)$' (where $i = 0$ or $i = 1$). If $v(\alpha_i) = 0$, then $v(F) = 0$ (for both $i = 0, 1$), therefore L-valuation v is also a falsifying L-valuation for sequence S.

The reasoning is analogous if F is a β-, κ- or ε-formula, and relies on the simple inequalities: $r(\beta) > r(\beta_0) + r(\beta_1)$, $r(\kappa) > r(\kappa_0)$, and finally $r(\varepsilon) > r(\varepsilon_{i0}) + r(\varepsilon_{i1})$ for $i \in \{0, 1\}$. We skip the details.

Now we have to consider a situation when $S \in \mathcal{F}$ and $r(S) = n$, but there is $k \geq 1$ formulas of rank n in sequence S. We reason by subinduction on k. The base step (for $k = 1$) has been proved above. Suppose $k > 1$ and S is of the form $S_1\,'\,F_1\,'\,S_2\,'\cdots'\,S_k\,'\,F_k\,'\,S_{k+1}$, where $F_1, \ldots, F_k$ are all the formulas of rank n and $S_1, S_2, \ldots, S_k, S_{k+1}$ are (possibly empty) sequences of formulas of rank lesser than n. We consider the form of formula F_1 and reason analogously as before. ■

As we can see now, a refutability property for L defined syntactically is a family of sequences which have falsifying L-valuations, that is, whose sets of terms are *semantically refutable in logic* L.

Now we may define:

Definition 14. (*RS-refutable sequence*) We say that a sequence, S, of formulas of $\mathcal{L}_{\mathsf{L}^+}$ is *RS-refutable in* $\mathbb{RS}^{\mathsf{L}}$ iff there is no successful decomposition diagram of S via the rules of $\mathbb{RS}^{\mathsf{L}}$.

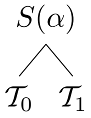
And finally:

Lemma 7. *Let $\mathcal{G}$ be a family of all finite and non-empty sequences S of formulas of $\mathcal{L}_{\mathsf{L}^+}$ which are RS-refutable in $\mathbb{RS}^{\mathsf{L}}$. $\mathcal{G}$ is a refutability property in* L.

Proof. We have to show that $\mathcal{G}$ satisfies each item of Definition 11.

Therefore let $S \in \mathcal{G}$. Observe that the tree containing only the root labelled with S is successful whenever S is fundamental. For this reason, if $S \in \mathcal{G}$, then S cannot be fundamental, thus item 1. of Definition 11 is satisfied.

Now we prove clause 2. Let $S(\alpha) \in \mathcal{G}$. Our aim is to show that then also $S(\alpha/\alpha_i) \in \mathcal{G}$ for $i = 0$ or $i = 1$. Therefore suppose that $S(\alpha/\alpha_i) \notin \mathcal{G}$ for both $i = 0, 1$. Then there are successful decomposition diagrams: $\mathcal{T}_0$ with '$S(\alpha/\alpha_0)$' in the root, and $\mathcal{T}_1$ with '$S(\alpha/\alpha_1)$' in the root. Then also the tree



is a successful decomposition diagram, and it shows that $S(\alpha) \notin \mathcal{G}$. We arrive at contradiction.

The reasoning goes analogously in the remaining cases and so we skip them. ■

Theorem 3. (Completeness of $\mathbb{RS}^{\mathsf{L}}$) *Let A be a formula of $\mathcal{L}_{\mathsf{L}}$. If A is* L*-valid, then A has a proof in* $\mathbb{RS}^{\mathsf{L}}$*.*

Proof. Suppose that A is not provable in $\mathbb{RS}^{\mathsf{L}}$. Then $\langle A\rangle$ is RS-refutable in $\mathbb{RS}^{\mathsf{L}}$. By Lemma 7, $\langle A\rangle$ belongs to a refutability property for L. By Lemma 6, there is an L-valuation v such that $v(A) = 0$. Thus, by contraposition, if A is L-valid, then A must be provable in $\mathbb{RS}^{\mathsf{L}}$. ■

5.1. Completeness of R–S Systems with Cut

Here we use the same technique and the proofs are analogous. We only state what is necessary.

Definition 15. (*cut-refutability property for* L) Let $\mathcal{F}$ be a family of finite sequences of formulas of $\mathcal{L}_{\mathsf{L}^+}$, the empty sequence included. We say that $\mathcal{F}$ is a *cut-refutability property for* L iff the following conditions are satisfied:

1. No $S \in \mathcal{F}$ is fundamental.
2. If $S(\alpha)(\overline{\alpha_i}) \in \mathcal{F}$, where $i \in \{0, 1\}$, then $S(\alpha/\alpha_j) \in \mathcal{F}$, where $j \in \{0, 1\}, j \neq i$.
3. If $S(\beta) \in \mathcal{F}$, then $S(\beta/\beta_0, \beta_1) \in \mathcal{F}$.
4. If $S(\kappa) \in \mathcal{F}$, then $S(\kappa/\kappa^*) \in \mathcal{F}$.
5. For $\mathsf{L} \in \{\mathsf{CPL}(\leftrightarrow), \mathsf{mbC}\}$, if $S(\varepsilon)(\overline{\varepsilon_{ik}}) \in \mathcal{F}$, where $i, k \in \{0, 1\}$, then $S(\varepsilon/\varepsilon_{jl}, \varepsilon_{jm}) \in \mathcal{F}$, where $j, l, m \in \{0, 1\}, i \neq j$ and $l \neq m$.
6. If $S' \, T \in \mathcal{F}$, then for each formula A of $\mathcal{L}_{\mathsf{L}^+}$, $S' \, A' \, T \in \mathcal{F}$ or $S' \, \neg A' \, T \in \mathcal{F}$.

Due to item 6. of the above definition, cut-refutability properties are infinite sets.

Lemma 8. (Counter-model existence theorem) *If S is an element of a cut-refutability property for* L*, then S has a falsifying* L*-valuation.*

For the purpose of proving this lemma we adopt the usual technique coupled with consistency properties (see [19], Section 3.6., pp. 52–57), but adjust it to our sequence-format. Namely, we will call a cut-refutability property $\mathcal{F}$ *subsequence closed*[7] iff every subsequence S^* of a sequence $S \in \mathcal{F}$ is already in $\mathcal{F}$.

[7] By *subsequence* of sequence S we mean any sequence that is created by deleting some elements of S without changing the order of the other elements. More formally, finite

We state without proof:

Fact 3. *Every cut-refutability property may be extended to one which is subsequence closed.*

By the way, observe that our subsequence closed cut-refutability properties are already of finite character. Finite sequences are sufficient for our aims, as we deal with the propositional level only. Now we will prove the counter-model existence lemma for subsequence closed properties, then Lemma 8 will follow as a corollary from this result and Fact 3.

Lemma 9. (Counter-model existence lemma for subsequence closed cut-refutability properties) *If S is an element of a cut-refutability property $\mathcal{F}$ for* L *and $\mathcal{F}$ is subsequence closed, then S has a falsifying* L*-valuation.*

Proof. Suppose that $\mathcal{F}$ is a subsequence closed cut-refutability property for L and assume that $S \in \mathcal{F}$. As we already know, if S is empty, then there exists a falsifying L-valuation for S, thus suppose it is not.

By and large, the proof will run analogously as the proof of Lemma 6. The base step is exactly the same.

We assume the induction hypothesis: for each sequence of rank less than n ($n > 0$) there exists a falsifying L-valuation. Let $S \in \mathcal{F}$ be such that $r(S) = n$. As in the proof of Lemma 6, we need to consider separately the case when there is exactly one formula F of rank n, and when there is $k > 1$ formulas of rank n.

Let $S = S_1 \,'\, F \,'\, S_2$ (this time we use the notation with concatenation), where F is the only formula in S of rank n. We skip the cases of F being a κ-formula and F being a β-formula.

Assume that F is an α-formula and let α_0, α_1 stand for the components of F. By item 6. of Definition 15:

(a) sequence $S^* = S_1 \,'\, \alpha \,'\, \overline{\alpha_0} \,'\, S_2$ belongs to $\mathcal{F}$ *or*

(b) sequence $S^\sharp = S_1 \,'\, \alpha \,'\, \alpha_0 \,'\, S_2$ belongs to $\mathcal{F}$

Obviously, $r(\alpha_0) < r(\alpha)$ and $r(\overline{\alpha_0}) < r(\alpha)$. If (a) holds, then by item 2. of Definition 15, $T = S_1 \,'\, \alpha_1 \,'\, \overline{\alpha_0} \,'\, S_2 \in \mathcal{F}$. Moreover, $r(\alpha_1) < n$, therefore $r(T) < n$, and as before, we arrive at the conclusion that there exists a falsifying L-valuation v for T which is also a falsifying L-valuation for S^*. Since v makes each term of S^* false, it also makes each term of S false.

Footnote 7 continued
n-term sequence is a function from $\{1, \ldots, n\}$ to a certain set (the set of terms of the sequence). A subsequence of a sequence is any restriction of this function.

Suppose that (b) is the case. This is the tricky part where we need the property of being subsequence closed. Consider the following sequence of literals: $S^{\sharp\sharp} = S_1{}'\ \alpha_0{}'\ S_2$. It is a subsequence of $S^{\sharp}$, so it belongs to $\mathcal{F}$. Moreover, $r(S^{\sharp\sharp}) < n$, thus the induction hypothesis applies to $S^{\sharp\sharp}$: there exists an L-valuation v falsifying $S^{\sharp\sharp}$. We have $v(\alpha_0) = 0$, and thus also $v(\alpha) = 0$, which shows that v is a falsifying L-valuation for $S^{\sharp}$. The same L-valuation falsifies S.

The situation is similar when F is an ε-formula. We use cut (that is item 6. of Definition 15) to introduce:

(a) sequence $S^* = S_1{}'\ \varepsilon{}'\ \overline{\varepsilon_{00}}{}'\ S_2$ belongs to $\mathcal{F}$ *or*

(b) sequence $S^{\sharp} = S_1{}'\ \varepsilon{}'\ \varepsilon_{00}{}'\ S_2$ belongs to $\mathcal{F}$

If (a) holds, then by item 5. of Definition 15 we arrive at the conclusion that there exists a falsifying L-valuation v for S^*, which falsifies also S.

If (b) holds, then we have to use item 6. of Definition 15 once again, to come to the conclusion that:

(c) sequence $S^* = S_1{}'\ \varepsilon{}'\ \varepsilon_{00}{}'\ \overline{\varepsilon_{01}}{}'\ S_2$ belongs to $\mathcal{F}$ *or*

(d) sequence $S^{\sharp} = S_1{}'\ \varepsilon{}'\ \varepsilon_{00}{}'\ \varepsilon_{01}{}'\ S_2$ belongs to $\mathcal{F}$

If (c) holds, then we reason as in the case (a), and (d) is the tricky part where we use the subsequence-closed property and the fact that $v(\varepsilon_{00}) = v(\varepsilon_{01}) = 0$ yields $v(\varepsilon) = 0$.

Now we reason by induction with respect to the number k of formulas of rank n in sequence S. The base case has been proved above and it is easy to see that the inductive part may be proved by exactly the same reasoning as that presented for the base step. Therefore we skip it. ■

Now we go back to:

Proof. of Lemma 8.

By Fact 3, every cut-refutability property for L may be extended to one which is subsequence closed, and by Lemma 9, if S is an element of a cut-refutability property $\mathcal{F}$ for L and $\mathcal{F}$ is subsequence closed, then S has a falsifying L-valuation. Therefore if S is an element of a cut-refutability property for L, then S has a falsifying L-valuation. ■

Analogously as before, we define the notion of RS-refutable sequence in $\mathbb{RS}^{\mathsf{L}}_{cut}$ and proceed to:

Lemma 10. *Let $\mathcal{G}$ be a family of all finite and non-empty sequences S of formulas of $\mathcal{L}_{\mathsf{L}^+}$ which are RS-refutable in $\mathbb{RS}^{\mathsf{L}}_{cut}$. $\mathcal{G}$ is a cut-refutability property for L.*

Proof. The reasoning is exactly as in the case of Lemma 7. Item 1. of Definition 15 is satisfied, as fundamental sequences are not RS-refutable. For the other items we show smoothly that if, *e.g.*, $S(\varepsilon/\varepsilon_{jn}, \varepsilon_{jm}) \notin \mathcal{F}$, then there exists a successful decomposition diagram that can be easily "extended" by adding the root with $S(\varepsilon)(\overline{\varepsilon_{ik}})$ to the top of it, and so finally we arrive at the conclusion that $S(\varepsilon)(\overline{\varepsilon_{ik}}) \notin \mathcal{F}$ (where i, j, k, n, m are suitably restricted).

We consider item 6. Let $S' \, T \in \mathcal{G}$. We want to show that then also for each formula A of $\mathcal{L}_{\mathsf{L}^+}$, $S' \, A' \, T \in \mathcal{G}$ or $S' \, \neg A' \, T \in \mathcal{G}$. Therefore suppose that for certain formula A, neither $S' \, A' \, T$ is in $\mathcal{G}$ nor $S' \, \neg A' \, T$ is in $\mathcal{G}$. By the definition of RS-refutable sequences it means that both: $S' \, A' \, T$ and $S' \, \neg A' \, T$ have successful decomposition diagrams. The two diagrams may be combined to obtain a successful decomposition diagram of sequence $S' \, T$. This is done exactly as in the proof of Lemma 7 for the case of α-formulas. ■

Theorem 4. (Completeness) *Let A be a formula of $\mathcal{L}_{\mathsf{L}}$. If A is L-valid, then A is provable in $\mathbb{RS}^{\mathsf{L}}_{cut}$.*

Proof. Suppose that it is not. Then $\langle A \rangle$ is RS-refutable. By Lemma 10, $\langle A \rangle$ belongs to a cut-refutability property. By Lemma 8, there is an L-valuation v such that $v(A) = 0$. Thus by contraposition, if A is L-valid, then A must be provable in $\mathbb{RS}^{\mathsf{L}}_{cut}$. ■

5.2. Completeness of Analytic Restrictions of R–S Systems with Cut

It was tempting to prove completeness of analytic R–S systems by the same technique which works for the unrestricted version. The following definition has been prepared for this occasion:

Definition 16. Let $\mathcal{F}$ be a family of finite sequences of formulas of $\mathcal{L}_{\mathsf{L}^+}$, the empty sequence included. We say that $\mathcal{F}$ is an *analytic cut-refutability property* iff clauses 1.-5. from Definition 15 are satisfied and

6*. If $S' \, T \in \mathcal{F}$, then for each formula A such that:

(i) $\{A, \neg A\} \cap Dec(S' \, T) \neq \emptyset$, and
(ii) neither A nor '$\neg A$' is a term of $S' \, T$,

it is the case that $S' \, A' \, T \in \mathcal{F}$ or $S' \, \neg A' \, T \in \mathcal{F}$.

In order to prove completeness we had to show that every analytic cut-refutability property is a subset of a refutability property. For this purpose we have developed a kind of Henkin-style construction; it turned out, however, that both the construction and the proof of its correctness get irrationally complicated, having nothing in common with the usual elegance of abstract properties. One hypothesis explaining why it is the case is that when the rules of deduction system are formulated in uniform notation they do not characterize the (classical) negation connective as such. Then adding a clause expressing cut, as an operation involving a formula and its negation, causes an "interference" which makes the proofs entangle in subcases of subcases.

Obviously, we may be wrong. But in order to prove completeness of the restricted analytic version of $\mathbb{RS}^{\mathsf{L}}_{cut}$ we decided to rely on a terminating proof-search procedure. Similar strategy has been adopted in [16,17], where the authors first prove completeness of **KE** by a smart argument referring **KE** to an axiomatic system (where the rule of cut is used to simulate *Modus Ponens*), and then prove completeness of the analytic restriction of **KE** by developing a suitable proof-search procedure.

5.2.1. Proof-Search Procedure for the Analytic Restriction of $\mathbb{RS}^{\mathsf{L}}_{cut}$ Let us stress, however, that this proof-search procedure is developed for the purpose of the completeness proof, and not for the purpose of "genuine" proof-search, since, first of all, the efficiency of the procedure must be poor.[8]

First of all, the proof-procedure aims at decreasing the rank of a sequence of formulas. Recall that the rank of a sequence is defined as the maximum of ranks of the formulas in the sequence. Therefore we start with a decomposition of a formula whose rank equals the rank of the analysed sequence. Let A stand for the leftmost such formula (there may be more than one). If A is a κ- or β-formula, then the things are simple: we apply the appropriate rule and check if there is another formula of the same rank in the sequence-conclusion. If A is an α- or ε-formula, then we check for the complement of its component: $\overline{\alpha_i}$ or $\overline{\varepsilon_{ik}}$. If there is one, then we apply the appropriate rule and, again, search for formulas of the same rank in the sequence-conclusion.

Now suppose that A is an α- or ε-formula but the suitable complement of its component is missing. We do what follows.

[8] As was rightly pointed out by one of the Reviewers, the cause of inefficiency is directly related to…the use of the rule of cut, since the whole sequence of formulas must be searched through before its application (see the description of the procedure).

- For α-formulas: check if there is a component of A in the analysed sequence. If there is one, then mark A as *analysed*, since the component "witnesses" A's falsity. If there is no component, then choose one of them, *e.g.* α_0, and apply R_{cut} introducing α_0 and $\overline{\alpha_0}$ as cut formulas. (Observe that this application of R_{cut} is analytic in the sense of Definition 10.) There are two sequences-conclusions. In the next step apply lR_α with respect to the sequence-conclusion with $\overline{\alpha_0}$, then formula A will be replaced with its component and thus one formula of the maximal rank disappears. In the second sequence-conclusion there is A and its component, thus mark A as *analysed*.
- For ε-formulas: we need two components of the forms ε_{i0}, ε_{i1} to witness A's falsity. If there is such a pair of components in the analysed sequence, then mark A as *analysed*. If there is exactly one component of a pair, *e.g.* ε_{00}, then apply R_{cut} with the second component of the pair and its complement, *e.g.* ε_{01} and $\overline{\varepsilon_{01}}$, as cut formulas. (Again, this application of R_{cut} is analytic.) Then in the sequence-conclusion with $\overline{\varepsilon_{01}}$ apply lR_ε, and in the sequence-conclusion with ε_{01} there is now a required pair of components, thus mark A as *analysed*.

  Finally, if there is no component of A in the analysed sequence, then choose one, *e.g.* ε_{00}, apply R_{cut} with ε_{00} and $\overline{\varepsilon_{00}}$ as cut formulas, and then apply R_{cut} again to the sequence-conclusion containing ε_{00}, introducing ε_{01} and $\overline{\varepsilon_{01}}$ as cut formulas. Both applications of R_{cut} are analytic. We obtain situations as above—in each of the resulting sequences A is either replaced with a pair of its components or marked as *analysed*.

On further steps of proof-search the formulas marked as *analysed* are ignored. By König's Lemma this kind of procedure terminates—this may be demonstrated with the use of rank of a sequence. When a decomposition diagram is finished, its leaves contain only formulas of rank 0 and/or those marked as *analysed*. If the leaves are fundamental, then we have a proof (a successful decomposition diagram). If the decomposition diagram is not successful, then we choose *e.g.* the leftmost leaf which is not a fundamental sequence and assign the logical value 0 to all its terms of rank 0. Simple reasoning by induction with respect to rank of formulas shows that every formula which is a term of a sequence on the same branch is false under each L-valuation which is an extension of the assignment. And this allows us to state:

Theorem 5. (Completeness of analytic restriction of $\mathbb{RS}^{\mathsf{L}}_{cut}$) *Let A be a formula of $\mathcal{L}_{\mathsf{L}}$. If A is* L*-valid, then A is provable in the analytic restriction of* $\mathbb{RS}^{\mathsf{L}}_{cut}$.

6. R–S Systems and the Logic of Questions

The inspiration to construct R–S systems presented in this paper came from research conducted in the framework of Inferential Erotetic Logic (IEL, for short; see [38] for a general introduction). The logic of questions IEL gave birth to two proof methods: the method of synthetic tableaux (see [36]) and the method of Socratic proofs, which is, roughly speaking, a method of transforming questions of certain formal languages concerning such important logical properties as validity. An *erotetic calculus* is a set of rules transforming such questions; at the same time, it constitutes a deduction system for the underlying logic. The method has been described for classical [12,37,39] and various non-classical logics [14,27,28,40].

The results presented in this paper were originally obtained in the framework of the method of Socratic proofs. We realised, however, that the format of R–S systems can be somewhat more general, hence the decision to take the proof-theoretical perspective and leave the erotetic aspects aside.[9]

However, here is something for the Readers familiar with the method of Socratic proofs and for those interested in erotetic reasoning. Using the uniform notation introduced in this paper, the erotetic calculi $\mathbb{E}^{\mathsf{L}}$ for logics $\mathsf{L} \in \{\mathsf{CPL}, \mathsf{CPL}(\leftrightarrow), \mathsf{CLuN}, \mathsf{CLuNs}, \mathsf{mbC}\}$ may be presented as follows.

$$\frac{?(\Phi \ ; \ \vdash S(\alpha) \ ; \ \Psi)}{?(\Phi \ ; \ \vdash S(\alpha/\alpha_0) \ ; \ \vdash S(\alpha/\alpha_1) \ ; \ \Psi)}\ R_\alpha$$

$$\frac{?(\Phi \ ; \ \vdash S(\beta) \ ; \ \Psi)}{?(\Phi \ ; \ \vdash S(\beta/\beta_0, \beta_1) \ ; \ \Psi)}\ R_\beta \qquad \frac{?(\Phi \ ; \ \vdash \ S(\kappa) \ ; \ \Psi)}{?(\Phi \ ; \ \vdash \ S(\kappa/\kappa_0) \ ; \ \Psi)}\ R_\kappa$$

$$\frac{?(\Phi \ ; \ \vdash S \ ' \ \varepsilon \ ' \ T \ ; \ \Psi)}{?(\Phi \ ; \ \vdash S \ ' \ \varepsilon_{00} \ ' \ \varepsilon_{01} \ ' \ T \ ; \ \vdash S \ ' \ \varepsilon_{10} \ ' \ \varepsilon_{11} \ ' \ T \ ; \ \Psi)}\ R_\varepsilon$$

[9] There is more to say abut the common denominator of the method of Socratic proofs and the R–S systems. In [22] the authors sketch an algorithmic procedure of generating a proof in an axiomatic system for CPL which is based on the use of the method of Socratic proofs. The result is constructive and leads, *int.al.*, to a conclusion that the method of Socratic proofs for CPL is polynomially simulated by the axiomatic system. However, originally, the research has been inspired by a similar work done in the framework of the R–S systems in [30]. As the authors of [22] admit, the result may be easily formulated with the use of the R–S machinery.

And the rules of $\mathbb{E}^{\mathsf{L}}_{cut}$ may be presented as follows.

$$\frac{?(\Phi\ ;\ \vdash S(\alpha)(\overline{\alpha_i})\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S(\alpha/\alpha_j)\ ;\ \Psi)}\ lR_\alpha \qquad \frac{?(\Phi\ ;\ \vdash S(\beta)\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S(\beta/\beta_0,\beta_1)\ ;\ \Psi)}\ R_\beta$$

where $i, j \in \{0,1\}, i \neq j$

$$\frac{?(\Phi\ ;\ \vdash S(\kappa)\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S(\kappa/\kappa_0)\ ;\ \Psi)}\ R_\kappa \qquad \frac{?(\Phi\ ;\ \vdash\ S\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ 'A\ '\ T\ ;\ \vdash S\ '\neg A\ '\ T\ ;\ \Psi)}\ R_{PB}$$

$$\frac{?(\Phi\ ;\ \vdash S(\varepsilon)(\overline{\varepsilon_{ik}})\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S(\varepsilon/\varepsilon_{jn},\varepsilon_{jm})\ ;\ \Psi)}\ lR_\varepsilon$$

where $i, j, k, n, m \in \{0,1\}, i \neq j$ and $n \neq m$.

7. Conclusions

In this paper we have presented two versions of Rasiowa–Sikorski deduction systems for the following logics: CPL, the propositional parts of paraconsistent logics CLuN and CLuNs, and mbC, which is the minimal logic of formal inconsistency. The first version of the R–S systems goes along the lines of R–S methodology, whereas the second one simulates the KE tableau calculus in the R–S framework. It turns out that the two frameworks may be combined with benefits, although the questions about the complexity of the obtained systems, especially their relative complexity, remain open. This subject may be further investigated.

One of the technical results of the presented research is the use of the so-called refutability properties—dual to consistency properties—in the completeness proof, especially the adjustment of this notion to the presented proof format. The technique of proving completeness by the use of the refutability properties seems promising and its usefulness will be examined in the future.

Funding The first and the third author were supported by the Polish National Science Center, Grant No. [2012/04/A/HS1/00715]. A part of this work concerning Classical Propositional Logic originated as a master thesis [24] defended by the second author in the

Department of Logic and Cognitive Science, Institute of Psychology, Adam Mickiewicz University.

References

[1] AGUDELO-AGUDELO, J.C., Translating Non-classical Logics into Classical Logic by Using Hidden Variables. *Logica Universalis* 11(2):205–224, 2017.

[2] BATENS, D., Paraconsistent extensional propositional logics. *Logique et Analyse* 90–91:195–234, 1980.

[3] BATENS, D., Inconsistency-adaptive logics. In E. Orłowska, (ed.), *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa*, Springer, Berlin, 1998, pp. 445–472.

[4] BATENS, D., and K. DE CLERCQ, A Rich Paraconsistent Extension of Full Positive Logic. *Logique et Analyse* 185–188:227–257, 2005.

[5] BATENS, D., K. DE CLERCQ, and N. KURTONINA, Embedding and Interpolation for Some Paralogics. The Propositional Case. *Reports on Mathematical Logic* 33:29–44, 1999.

[6] BATENS, D., and J. MEHEUS, A Tableau Method for Inconsistency-Adaptive Logics. In R. Dyckhoff, (ed.), *Automated Reasoning with Analytic Tableaux and Related Methods, Lecture Notes in Artificial Intelligence*, Springer, Berlin, 2000, pp. 127–142.

[7] BATENS, D., and J. MEHEUS, Shortcuts and Dynamic Marking in the Tableau Method for Adaptive Logics. *Studia Logica* 69:221–248, 2001.

[8] BOOLOS, G., Don't Eliminate Cut. *Journal of Philosophical Logic* 13(4):373–378, 1984.

[9] CALEIRO, C., J. MARCOS, and M. VOLPE, Bivalent semantics, generalized compositionality and analytic classic-like tableaux for finite-valued logics. *Theoretical Computer Science* 603:84–110, 2015.

[10] CARNIELLI, W.A., and M.E. CONIGLIO, Logics of Formal Inconsistency. In F. Guenthner, and D.M. Gabbay, (eds.), *Handbook of Philosophical Logic, vol. 14*, Springer, Berlin, 2013, pp. 1–93.

[11] CARNIELLI, W.A., and J. MARCOS, A taxonomy of **C**-systems. In I.M.L. D'Ottaviano, W.A. Carnielli, and M.E. Coniglio, (eds.), *Paraconsistency—The Logical Way to the Inconsistent*, Marcel Dekker, 2000, pp. 1–94.

[12] CHLEBOWSKI, S.Z., *Canonical and Dual Erotetic Calculi for First-Order Logic*. Ph.D. Thesis, Adam Mickiewicz University, Poznań, 2018. (Unpublished manuscript, previously referred to as "The Method of Socratic Proofs for Classical Logic and Some Non-Classical Logics").

[13] CHLEBOWSKI, S.Z., A. GAJDA, and M. URBAŃSKI, Abductive Question–Answer System for the Minimal Logic of Formal Inconsistency mbC. (Unpublished manuscript).

[14] CHLEBOWSKI, S.Z., and D. LESZCZYŃSKA-JASION, Dual Erotetic Calculi and the Minimal LFI. *Studia Logica* 103(6):1245–1278, 2015.

[15] CONIGLIO, M.E., and T.G. RODRIGUES, Some investigations on mbC and mCi. In C.A. Mortari, (ed.), *Tópicos de lógicas não clássicas*, NEL/UFSC, 2014, pp. 11–70.

[16] D'AGOSTINO, M., Are tableaux an improvement on truth-tables? *Journal of Logic, Language and Information* 1(3):235–252, 1992.

[17] D'AGOSTINO, M., and M. MONDADORI, The Taming of the Cut. Classical Refutations with Analytic Cut. *Journal of Logic and Computation* 4(3):285–319, 1994.

[18] Fitting, M., *Proof Methods for Modal and Intuitionistic Logics.* Springer, Netherlands, 1983.

[19] Fitting, M., *First-order Logic and Automated Theorem Proving.* Springer, Berlin, 1990.

[20] Golińska-Pilarek, J., T. Huuskonen, and E. Muñoz-Velasco, Relational dual tableau decision procedures and their applications to modal and intuitionistic logics. *Annals of Pure and Applied Logic* 165(2):409–427, 2014.

[21] Golińska-Pilarek, J., and E. Orłowska. Tableaux and dual tableaux: transformation of proofs. *Studia Logica* 85:283–302, 2007.

[22] Grzelak, A., and D. Leszczyńska-Jasion, Automatic proof generation in an axiomatic system for CPL by means of the method of Socratic proofs. *Logic Journal of the IGPL* 26(1):109–148, 2018.

[23] Hähnle, R., Tableaux and Related Methods. In A. Robinson, and A. Voronkov, (eds.), *Handbook of Automated Reasoning*, Chapter 3, Elsevier Science Publishers, Amsterdam, 2001, pp. 101–175.

[24] Ignaszak, M., *Dual erotetic version of system KE.* Master's Thesis, Department of Logic and Cognitive Science, Institute of Psychology, Adam Mickiewicz University, 2017.

[25] Konikowska, B., Rasiowa–Sikorski deduction system: a handy tool for Computer Science logic. In *Proceedings WADT98, Springer Lecture Notes in Computer Science*, volume 1589, Springer, Berlin, 1999, pp. 183–197.

[26] Konikowska, B., Rasiowa–Sikorski deduction systems in computer science applications. *Theoretical Computer Science* 286(2):323–366, 2002.

[27] Leszczyńska-Jasion, D., *The Method of Socratic Proofs for Normal Modal Propositional Logics.* Adam Mickiewicz University Press, Poznań, 2007.

[28] Leszczyńska-Jasion, D., The Method of Socratic Proofs for Modal Propositional Logics: K5, S4.2, S4.3, S4M, S4F, S4R and G. *Studia Logica* 89(3):371–405, 2008.

[29] Neto, A.G.S.S., and M. Finger, Effective prover for minimal inconsistency logic. In M. Bramer, (ed.), *IFIP International Federation for Information Processing*, Springer, Berlin, 2006, pp. 465–474.

[30] Orłowska, E., and J. Golińska-Pilarek, *Dual Tableaux: Foundations, Methodology, Case Studies*, volume 33 of *Trends in Logic.* Springer, Dordrecht, 2011.

[31] Pudlák, P., The Lengths of Proofs. In S.R. Buss, (ed.), *Handbook of Proof Theory*, chapter VIII, Elsevier, 1998, pp. 547–637.

[32] Rasiowa, H., and R. Sikorski, On the Gentzen theorem. *Fundamenta Mathematicae* 48:57–69, 1960.

[33] Rasiowa, H., and R. Sikorski, *The Mathematics of Metamathematics.* Polish Scientific Publishers, Warsaw, 1963.

[34] Smullyan, R.M., *First-Order Logic.* Springer, Berlin, 1968.

[35] Smullyan, R.M., *A Beginner's Guide to Mathematical Logic.* Dover Books on Mathematics. Dover Publications, 2014.

[36] Urbański, M., *Tabele syntetyczne a logika pytań.* Wydawnictwo UMCS, Lublin, 2002.

[37] Wiśniewski, A., Socratic Proofs. *Journal of Philosophical Logic* 33(3):299–326, 2004.

[38] Wiśniewski, A., *Questions, Inferences, and Scenarios.* College Publications, London, 2013.

[39] Wiśniewski, A., and V. Shangin, Socratic proofs for quantifiers. *Journal of Philosophical Logic* 35(2):147–178, 2006.

[40] Wiśniewski, A., G. Vanackere, and D. Leszczyńska, Socratic Proofs and Paraconsistency: A Case Study. *Studia Logica* 80(2-3):433–468, 2005.

D. Leszczyńska-Jasion, M. Ignaszak, S. Chlebowski
Department of Logic and Cognitive Science, Institute of Psychology
Adam Mickiewicz University
Poznań
Poland
Dorota.Leszczynska@amu.edu.pl