Logique & Analyse **185–188** (2004), 319–333

HOW TO SYNTHESIZE A PARACONSISTENT NEGATION: THE CASE OF CLUN

MARIUSZ URBAŃSKI*

Abstract

The aim of this paper is to apply synthetic tableaux method (STM) to the paraconsistent logic CLuN, developed by Diderik Batens. Soundness and completeness of STM with respect to CLuN semantics are proved. It is also shown how to interpret CLuN negation in terms of relations represented by the square of oppositions of traditional syllogistic.

1. *Introduction*

The logic CLuN, developed by Diderik Batens, is a first-order paraconsistent logic. It is obtained by dropping from the classical semantics the following requirement for negation:

(*) if A is true, then $\sim A$ is false;

and by keeping its converse:

(**) if A is false, then $\sim A$ is true.

Thus, CLuN allows for gluts with respect to negation. As a result, formulas of the form $\sim A$, and no other formulas, may be true independently of the truth value of their subformulas. This comes to the fact that CLuN negation is not truth-functional (cf. e.g. Batens (1986), (1989), (2003) for more details).

Synthetic tableaux method (STM) is a semantically motivated decision method based on direct reasoning. The main idea underlying STM is to

---

*The author was supported by Foundation for Polish Science. The research for this paper also benefited from a bilateral scientific exchange project funded by the Ministry of the Flemish Community (project BIL01/80) and the Polish State Committee for Scientific Research.

solve *via* a tableau the following problem: which (compound) formulas are "synthesizable" (can be derived from the simpler ones) on the basis of certain sets of (atomic) formulas (cf. Urbański (2001a), (2002a)). The aim of this paper is to apply STM as a proof method for CLuN (although in this paper we restrict ourselves to its propositional part, we will be using the name CLuN).

2. *CLuN: language and semantics*

The language J of CLuN is an extension of the language of Classical Propositional Calculus. We will deal here with the version with $\neg$ (classical negation), $\wedge$ (classical conjunction), $\vee$ (classical disjunction), $\rightarrow$ (classical implication) and $\sim$ (paraconsistent negation) as primitive connectives. The notion of well-formed formula (wff for short) is defined as usual. We do not restrict the application of negation signs, thus allowing for the possibility that negation of one type occurs within the scope of the negation of the other type. The set of all the wffs of J will be referred to as $Form$. We will be using $p, q, r, p_1, \ldots$ for propositional variables, $\varphi, \phi, \varphi_1, \ldots$ as metavariables for them and $A, B, C, \ldots$ as metavariables for wffs.

One characteristic feature of CLuN-semantics is that the truth values of paraconsistently negated formulas (i.e., wffs of the form $\sim A$) are assigned directly. The assignment v' assigns 1 (Truth) or 0 (Falsehood) to propositional variables whereas the assignment v'' assigns 1 or 0 to the formulas of the form $\sim A$. Assume that Var is the set of all the propositional variables of J and that $Form^{\sim} = \{\sim A : A \in Form\}$. The assignment functions fulfil the following conditions:

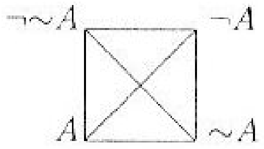
(A.1) $\mathsf{v}' : Var \longmapsto \{1, 0\}$
(A.2) $\mathsf{v}'' : Form^{\sim} \longmapsto \{1, 0\}$

A function $\mathsf{v} : Form \longmapsto \{1, 0\}$ is a CLuN-valuation iff:

(V.1) $\mathsf{v}(A) = \mathsf{v}'(A)$, if $A \in Var$
(V.2) $\mathsf{v}(\neg A) = 1$ iff $\mathsf{v}(A) = 0$
(V.3) $\mathsf{v}(\sim A) = 1$ iff $\mathsf{v}(A) = 0$ or $\mathsf{v}''(\sim A) = 1$
(V.4) $\mathsf{v}(A \wedge B) = 1$ iff $\mathsf{v}(A) = 1$ and $\mathsf{v}(B) = 1$
(V.5) $\mathsf{v}(A \vee B) = 1$ iff $\mathsf{v}(A) = 1$ or $\mathsf{v}(B) = 1$
(V.6) $\mathsf{v}(A \rightarrow B) = 1$ iff $\mathsf{v}(A) = 0$ or $\mathsf{v}(B) = 1$

Other semantical notions are defined as usual. In particular, a formula A is CLuN-valid iff for every CLuN-valuation v, $\mathsf{v}(A) = 1$.

One can easily observe that, although the semantics for CLuN is defined by means of two separate assignments, it is not the case that the truth values of the formulas A, $\neg A$ and $\sim A$ are unrelated to each other. Not surprisingly, they can be adequately described within the framework of Aristotle's theory of oppositions[1] and depicted in the following CLuN-version of the square of oppositions:



The relations that hold between pairs of connected formulas are exactly the same as in the classical square of oppositions:

(CT) $\neg\sim A$ and $\neg A$ are *contrariae*;
(SC) A and $\sim A$ are *subcontrariae*;
(CR) A and $\neg A$ as well as $\neg\sim A$ and $\sim A$ are *contradictoriae*;
(SA) A is *subalternae* to $\neg\sim A$ as well as $\sim A$ is *subalternae* to $\neg A$.

This means, in particular, that under the very same valuation:

(CT$'$) $\neg\sim A$ and $\neg A$ can be both false, but cannot be both true;
(SC$'$) A and $\sim A$ can both be true, but cannot be both false.

It is worth noticing that (SA) is equivalent to the following:

(SA$'$) A is entailed by $\neg\sim A$ as well as $\sim A$ is entailed by $\neg A$.

In what follows we will be using the concept of *signed formulas*: where $A \in Form$, TA and FA are signed formulas (T and F will be referred to as *truth-signs*). We will refer to them as to 'formulas' in cases where no ambiguity can arise. We will use &, # as variables for truth-signs, if needed.

Truth-signs do not belong to the vocabulary of the language J, so the truth values of signed formulas are not determined by CLuN-valuations. Nevertheless, the truth value of a signed formula $\#A$ (where # is any of T, F) is dependent upon the truth value of the formula A under a certain valuation, so we will speak of the truth value of a formula $\#A$ *with respect* to that valuation. The definition of this notion is given by the following table (in the leftmost column there is indicated the truth value of a formula A under a

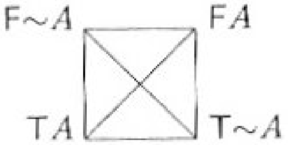
---

[1] Cf. Bocheński (1951), or Łukasiewicz (1951).

valuation in question):

A	TA	FA
1	1	0
0	0	1

Rewritten with signed formulas the above square of oppositions will look like this:



This version makes the relations between a formula and its CLuN-negation more explicit.

3. *Synthetic Tableaux*

We will make use of the following notion of a subformula of a given wff. If A is a propositional variable, then it has no proper subformulas at all. If A is a formula of the form '$\neg B$' or of the form '$\sim B$', then B is a *proper subformula* of A and, at the same moment, the only *immediate subformula* of A. If A is of the form 'B^*C' (where '*' stands for any of the binary connectives), then both B, C are *proper subformulas* of A and the only *immediate subformulas* of A. If C is a proper subformula of B and B is a proper subformula of A, then C is a proper subformula of A. A formula B is a *subformula* of a formula A iff B is a proper subformula of A or $A = B$.

Definition 1: *A finite sequence* s $= s_1, ..., s_n$ *of signed wffs is a synthetic inference of* $\#A$ *iff:*

(1) *every term of* s *is a signed subformula of* A;
(2) s_1 *is a signed propositional variable;*
(3) s_n *is* $\#A$;
(4) *for every* s_g *(where* $g = 1, ..., n$*), either* s_g *is a signed propositional variable, or* s_g *is derivable via CLuN-rules on the basis of a certain set of wffs such that each element of this set precedes* s_g *in* s;
(5) *for every* s_g *(where* $g = 1, ..., n$*) of* s *the following holds:*
    (a) *if* s_g *is a signed propositional variable, then none of* Tφ, Fφ *occurs at any other place in* s;

(b) *if s_g is a signed paraconsistently negated wff (that is, a wff of the form $\sim B$, preceded by a truth sign), then none of $\mathsf{T}\sim B$, $\mathsf{F}\sim B$ occurs at any other place in* s;

Thus a synthetic inference of a signed formula $\#A$ is a finite sequence of signed subformulas of it, which begins with some signed propositional variable and ends with $\#A$ itself. Moreover, every (signed) propositional variable occurs as a term in s only once, no matter of truth-signs (the same pertains to the (signed) paraconsistently negated wffs in s), and every formula which is not a (signed) propositional variable is derivable form some earlier formula(s) of s by means of CLuN-rules.

We define the set of CLuN-rules as the union of the following sets of CL-rules and of N-rules:

CL-rules:
$\mathsf{T}A/\mathsf{F}\neg A$ $\quad$ $\mathsf{F}A/\mathsf{T}\neg A$

$\mathsf{T}A/\mathsf{T}A \vee B$ $\quad$ $\mathsf{T}B/\mathsf{T}A \vee B$ $\quad$ $\mathsf{F}A, \mathsf{F}B/\mathsf{F}A \vee B$

$\mathsf{F}A/\mathsf{F}A \wedge B$ $\quad$ $\mathsf{F}B/\mathsf{F}A \wedge B$ $\quad$ $\mathsf{T}A, \mathsf{T}B/\mathsf{T}A \wedge B$

$\mathsf{F}A/\mathsf{T}A \to B$ $\quad$ $\mathsf{T}B/\mathsf{T}A \to B$ $\quad$ $\mathsf{T}A, \mathsf{F}B/\mathsf{F}A \to B$

N-rules:

rule NT1	rule NT2	rule NF
$\mathsf{T}A/\mathsf{T}\sim A$	$\mathsf{T}A/\mathsf{F}\sim A$	$\mathsf{F}A/\mathsf{T}\sim A$

All the CL-rules are obviously sound, as well as the rule NF (recall, that the subalternation relation that holds between the premise and the conclusion of this rule is, in fact, entailment). The rules NT1 and NT2, however, are not.

Our main notion is given by the following definition:

Definition 2: Let A be an unsigned wff and let $\#$, $\&$ be distinct truth-signs. A family Ω of finite sequences of signed formulas is a synthetic tableau for A iff every element of Ω is a synthetic inference of $\mathsf{T}A$ or of $\mathsf{F}A$, there exists a propositional variable φ such that the first element of every sequence in Ω is $\mathsf{T}\varphi$ or $\mathsf{F}\varphi$, and for every sequence $\mathsf{s} = s_1, ..., s_n$ in Ω the following hold:

(1) *if s_i (where $i = 1, ..., n$) is a signed propositional variable $\#\varphi$, then:*
  (a) *there exists in Ω a sequence s^* such that s^*_i is $\&\varphi$ and, if $i > 1$, then s and s^* do not differ to the level of their $i-1$th terms;*

   (b) *if* $i > 1$, *then for every sequence* s^* *in* Ω *such that* s *and* s^* *do not differ to the level of their* $i - 1$*th terms,* s_i^* *is* $\mathsf{T}\varphi$ *or* $\mathsf{F}\varphi$*;*

(2) *if* s_i *(where* $i = 2, ..., n$*) is of the form* $\#{\sim}B$ *and there exists* s_h *(*$h < i$*) such that* s_h *is of the form* $\mathsf{T}B$*, then:*
   (a) Ω *contains a sequence* s^* *such that* s *and* s^* *do not differ to the level of their* $i - 1$*th terms and* s_i^* *is* $\&{\sim}B$*;*
   (b) *for every sequence* s^* *in* Ω *such that* s *and* s^* *do not differ to the level of their* $i - 1$*th terms,* s_i^* *is* $\#{\sim}B$ *or* $\&{\sim}B$*.*

Thus, a synthetic tableau Ω for a formula A is a set of interconnected synthetic inferences (we will refer to them as to *branches* of Ω) of $\mathsf{T}A$ or of $\mathsf{F}A$. Every sequence in Ω begins with a fixed propositional variable, preceded with a truth-sign.

The clause (1) of the above definition expresses a 'fair branching' condition of a synthetic tableau with respect to propositional variables. If the i-th element of a certain synthetic inference s in Ω is a signed propositional variable $\#\phi$, then Ω contains synthetic inference s^* such that it does not differ from s to the level of their $i - 1$th terms and whose i-th term is $\&\phi$. Moreover, if a synthetic inference s in Ω has a signed propositional variable $\#\phi$ as its i-th term ($i > 1$), then each synthetic inference in Ω which is identical with s to the level of their $i - 1$th terms has as its i-th term either $\mathsf{T}\phi$, or $\mathsf{F}\phi$.

The clause (2), in turn, expresses a 'fair branching' condition with respect to the formulas of the form $\#{\sim}B$ in presence of $\mathsf{T}B$. According to this clause, any application of the rule NT1 (resp. NT2) is accompanied by an application of the rule NT2 (resp. NT1). Thus as soon as $\#{\sim}B$ is derived as i-th term of a branch s (by NT1 or NT2) the tableau is splitted into s and s^* such that s and s^* do not differ to the level of their $i - 1$th terms and s_i^* is $\&{\sim}B$. Moreover, if a synthetic inference s in Ω has $\#{\sim}B$ as its i-th term ($i > 1$), then each synthetic inference in Ω which is identical with s to the level of their $i - 1$th terms has as its i-th term either $\mathsf{T}{\sim}B$, or $\mathsf{F}{\sim}B$.

Clauses (1) and (2) taken together warrant the soundness of the method. Signed propositional variables are the only formulas that are introduced into synthetic inferences not as derived ones; on the other hand, formulas of the form $\#{\sim}B$ may be introduced by means of unsound rules. Fair branching guarantees that in both cases splitting of a tableau preserves soundness. These clauses form a kind of a cut rule. It is, of course, not an inferential rule; it is a tree-construction rule. Moreover, this cut is very restricted: it can be applied only to propositional variables or to the paraconsistently negated formulas (provided that they are subformulas of the initial formula). Therefore, the restrictions here are even stronger than in case of Smullyan's analytic cut.

Another point is, that if a formula A is of one of the 'branching-forcing' type (that is, it is either a propositional variable or it is of the form ${\sim}B$), the splitting of a tableau on TA and FA can be done only once (cf. clause (5) of definition 1). Thus, no branch of a tableau can contain the very same wff preceded with T at one place and with F at another.

Intuitively, a synthetic inference of a formula $\#A$ is a derivation of $\#A$ on the basis of a certain set of signed propositional variables. It can be proved (and we will prove this in the next section) that if X is a set made up off all the terms of a certain synthetic inference s, then there exists a CLuN-valuation v such that all the elements of X are true with respect to v (we use the phrase "true with respect to v" (and not just "true under v") as all the terms of s (and thus all the elements of X) are signed formulas). In view of 'fair branching' conditions, one can expect the following soundness-completeness theorem: a formula A is CLuN-valid iff there exists a synthetic tableau for A such that every path of this tableau leads to TA. We will prove this theorem as well.

4. *Soundness and completeness*

By a *degree* of a formula A (in symbols: $\deg(A)$) we mean the number of occurrences of connectives in A. Thus the degree of a propositional variable is 0, the degree of $\sim A$ as well as of $\neg A$ is $\deg(A) + 1$, the degree of $A^* B$ (where * stands for any of the binary connectives) is $\deg(A) + \deg(B) + 1$.

In order to proceed we need the following lemma:

Lemma 1: Let s *be a synthetic inference of a formula $\#A$. Let X be the set made up of all the terms of* s. *Let Θ be a subset of X made up of all the signed propositional variables in X. Then every formula in X is derivable via CLuN-rules on the basis of the set Θ.*

The proof is similar to that in Urbański (2002b). We will implicitly use this lemma to prove the following:

Theorem 1: Let s *be a synthetic inference of a (signed) formula A. Let X be the set made up of all the terms of* s. *Then there exists a valuation* v *such that all the elements of X are true with respect to* v.

Note, that by the definition of the synthetic inference, the following hold:

(1) if T$\varphi \in X$, then F$\varphi \notin X$ and if F$\varphi \in X$, then T$\varphi \notin X$;

(2) if F$\sim A \in X$, then T$A \in X$ and F$A \notin X$

Proof:
Let v be a valuation that is determined by the assignments v′, v″ such that they fulfil the following conditions:

(*) for every propositional variable φ:
 (i) if $\mathsf{T}\varphi \in X$, then $\mathsf{v}'(\varphi) = 1$
 (ii) if $\mathsf{F}\varphi \in X$, then $\mathsf{v}'(\varphi) = 0$
 (iii) if neither $\mathsf{T}\varphi$ nor $\mathsf{F}\varphi$ is an element of X then $\mathsf{v}'(\varphi) = 0$

(**) for every formula of the form $\sim A$:
 (i) if $\mathsf{T}{\sim}A \in X$, then $\mathsf{v}''({\sim}A) = 1$
 (ii) if $\mathsf{F}{\sim}A \in X$, then $\mathsf{v}''({\sim}A) = 0$
 (iii) if neither $\mathsf{T}{\sim}A$ nor $\mathsf{F}{\sim}A$ is an element of X, then $\mathsf{v}''({\sim}A) = 0$[2]

Consider the truth value of a formula $\#B$ in X with respect to v.

1. If $\deg(B) = 0$, then B is a propositional variable, and, by condition (*), $\#B$ is true with respect to v.

Suppose that theorem holds for all the formulas in X of the degree $k < n$. We will show that it holds for the formulas of the degree n as well.

2. Suppose that $\deg(B) = n$ $(n > 0)$. In this case $\#B$ is a derived compound formula. There are the following possibilities:
(a) $B = \neg C$
In this case:
- if $\mathsf{T}\neg C \in X$, then also $\mathsf{F}C \in X$; as $\deg(C) < \deg(\neg C)$, $\mathsf{F}C$ is true with respect to v; this means that $\mathsf{v}(C) = 0$; thus $\mathsf{v}(\neg C) = 1$ and $\mathsf{T}\neg C$ is true with respect to v;
- if $\mathsf{F}\neg C \in X$, then also $\mathsf{T}C \in X$; as $\deg(C) < \deg(\neg C)$, $\mathsf{T}C$ is true with respect to v; this means that $\mathsf{v}(C) = 1$; thus $\mathsf{v}(\neg C) = 0$ and $\mathsf{F}\neg C$ is true with respect to v;
(b) $B = {\sim}C$
- if $\mathsf{T}{\sim}C \in X$, then $\mathsf{v}''({\sim}C) = 1$; this means that $\mathsf{v}({\sim}C) = 1$ and thus $\mathsf{T}{\sim}C$ is true with respect to v[3];

---

[2] Obviously, the only role of the clause (iii) of condition (*) and the clause (iii) of condition (**) is to make the valuation v, in a sense, "complete".

[3] There are, in fact, two possibilities. If $\mathsf{T}{\sim}C$ is introduced *via* rule NT1, we reason as above. If, in turn, $\mathsf{T}{\sim}C$ is introduced *via* rule NF, then, as $\mathsf{F}C$ is true with respect to v, $\mathsf{v}(C) = 0$, $\mathsf{v}({\sim}C) = 1$ and $\mathsf{T}{\sim}C$ is true with respect to v (thus in this case $\mathsf{T}{\sim}C$ receives its truth-value because of both v′ and v″).

- if $\mathsf{F}{\sim}C \in X$, then also $\mathsf{T}C \in X$; as $\deg(C) < \deg({\sim}C)$, $\mathsf{T}C$ is true with respect to v; this means that $\mathsf{v}(C) = 1$; as $\mathsf{v}''({\sim}C) = 0$, then $\mathsf{v}({\sim}C) = 0$ and $\mathsf{F}{\sim}C$ is true with respect to v;
(c) $B = C \wedge D$
- if $\mathsf{T}C \wedge D \in X$, then also $\mathsf{T}C \in X$ and $\mathsf{T}D \in X$; as $\deg(C), \deg(D) < \deg(C \wedge D)$, $\mathsf{T}C$ and $\mathsf{T}D$ are true with respect to v; this means that $\mathsf{v}(C) = \mathsf{v}(D) = 1$; thus $\mathsf{v}(C \wedge D) = 1$ and $\mathsf{T}C \wedge D$ is true with respect to v;
- if $\mathsf{F}C \wedge D \in X$, then either $\mathsf{F}C \in X$ or $\mathsf{F}D \in X$; as $\deg(C), \deg(D) < \deg(C \wedge D)$, either $\mathsf{F}C$ or $\mathsf{F}D$ are true with respect to v; this means that $\mathsf{v}(C) = 0$ or $\mathsf{v}(D) = 0$; thus $\mathsf{v}(C \wedge D) = 0$ and $\mathsf{F}C \wedge D$ is true with respect to v;
In the remaining cases the reasoning is similar.■

Theorem 2: *There exists a valuation* v *such that the signed formula* $\#A$ *is true with respect to* v *iff there exists a synthetic inference of* $\#A$.

Proof:
(⇒) The proof of this part of theorem 2 consists in describing a method of construction of a synthetic inference of $\#A$ and is analogous to the one that can be found in Urbański (2002b).

(⇐) By theorem 1.■

One can also prove the following:

Lemma 2: *For every formula* A *of the language* J, *there exists a synthetic tableau for* A.

In the completeness proof of STM for CLuN, we will use the notion of a minimal error point of a synthetic inference:

Lemma 3: *Let a sequence* $\mathsf{s} = s_1, \ldots, s_n$ *be a synthetic inference of a certain formula* $\#A$ *and let* v *be a valuation such that not all the terms of* s *are true with respect to* v. *Then there exists an index* k $(k = 1, \ldots, n)$ *such that:*
(i) $s_k = \&B$, *where* B *is a propositional variable or* $B = {\sim}C$;
(ii) s_k *is false with respect to* v;
(iii) *there is no* $i < k$ *such that* s_i *is false with respect to* v.
We call k *the* minimal error point[4] *of* s *with respect to* v.

---

[4] This notion is due to A. Wiśniewski (cf. Wiśniewski (2003)).

Proof:
If not all the terms of the sequence s are true with respect to v, then one of them must occur in s with the lowest index — let it be k. Thus k is an index such that s_k is false with respect to v and there is no $i < k$ such that s_i is false with respect to v. Therefore, all the terms that precede s_k in s (if any) are true with respect to v. Moreover, one of the following holds:

(a) s_k is a signed propositional variable, or
(b) s_k is obtained by means of the CLuN-rules; there are two possibilities:
  - s_k is obtained by application of one of the CL-rules or by the rule NF; as these rules are sound and all the terms that precede s_k in s are true with respect to v, s_k must be true with respect to v as well;
  - s_k is obtained by application of the rule NT1 or NT2; in this case s_k is of the form $\& \sim C$.■

Now we are in a position to prove:

Theorem 3: A formula A is CLuN-valid iff there exists a synthetic tableau for A such that every path of it leads to TA.

Proof:
($\Rightarrow$) Let A be a CLuN-valid formula and let Ω be a synthetic tableau for A such that at least one path s of Ω leads to FA. Let X be the set made up of all the terms of s. By the theorem 1 there exists a valuation v such that all the elements of X (FA included) are true with respect to v. Thus $\mathsf{v}(A) = 0$ and A is not CLuN-valid. We arrive at a contradiction.

($\Leftarrow$)[5] Assume that there exists a synthetic tableau Ω for a formula A such that every path of Ω leads to TA. Moreover, assume (for an indirect proof) that A is not CLuN-valid. All the terms of paths of Ω are signed subformulas of A, and hence Ω is a finite set.
Since A is not CLuN-valid, there exists a certain valuation v such that A is false under v, that is, FA is true with respect to v. Since every path of Ω leads to TA, then for every path the set made up of all of its terms must contain some signed formulas that are false with respect to v. Therefore, by lemma 3, for every path of Ω there exists a minimal error point with respect to the valuation v. Let ω be the set of all the minimal error points of the paths of Ω with respect to the valuation v (so, ω is a set of indices, *i.e.* positive integers).
The set ω is finite and thus must have a maximal element, that is, there exists

---

[5] This part of the proof is based on the idea that comes from Wiśniewski (2003).

an index k such that for any j in ω, $j \leq k$. Let s be a path of Ω such that its minimal error point is the maximal one, that is, k. By lemma 3, s_k is either a signed propositional variable or is of the form $\&{\sim}C$, s_k is false with respect to v and either s_k is the first term of s, or every term that precedes s_k in s is true with respect to v.

Let $\#$, $\&$ be distinct truth-signs. If s_k is the first term of s, then there is a path s* of Ω such that, if $s_k = \#\phi$, then $s^*_k = \&\phi$. Therefore, s^*_k is bound to be true with respect to v.
If s_k is not the first term of s, then there exists path s* of Ω such that for every s_d which precedes s_k in s, $s_d = s^*_d$ (remember that every term of s which precedes s_k is true with respect to the valuation v) and, either, if $s_k = \#\phi$, then $s^*_k = \&\phi$, or, if $s_k = \#{\sim}C$, then $s^*_k = \&{\sim}C$. Therefore, s^*_k is bound to be true with respect to v.
By lemma 3, for every path of Ω there exists a minimal error point with respect to the valuation v, so there exists also such a minimal error point of the path s*. According to what has been shown above, the minimal error point of s* must be greater than the minimal error point of s, that is, it must be greater than the maximal element of ω. We arrive at a contradiction. Therefore for at least one path in $\Omega$the set made up of all of its terms must contain signed formulas that are all true with respect to v. Since every path of Ω leads to TA, then TA is true with respect to v and thus A itself is true under v.■

5. *Some examples*.

Consider the following variants of the law of contradiction[6] and the problem of their CLuN-validity.

The formula '$p \wedge \neg p$' expresses classical inconsistency (p and '$\neg p$' are *contradictoriae*) and is false under every CLuN-valuation. Thus '${\sim}(p \wedge \neg p)$' is a CLuN-valid formula (cf. example 1).

---

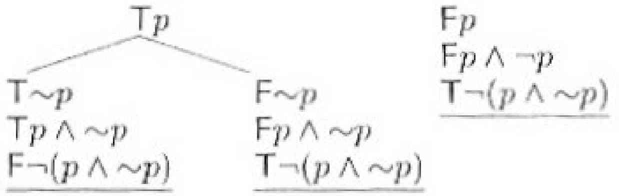
[6] We call all the formulas of examples 1–4 "variants of the law of contradiction", although for some of them a better name can be devised in view of the abovementioned CLuN-version of the square of oppositions.

Example 1: a synthetic tableau for '${\sim}(p \wedge \neg p)$'[7]

$$\begin{array}{ll} \mathsf{T}p & \mathsf{F}p \\ \mathsf{F}\neg p & \mathsf{F}p \wedge \neg p \\ \mathsf{F}p \wedge \neg p & \underline{\mathsf{T}{\sim}(p \wedge \neg p)} \\ \underline{\mathsf{T}{\sim}(p \wedge \neg p)} & \end{array}$$

The meaning of the formula '$\neg(p \wedge {\sim}p)$' is that p and '${\sim}p$' cannot be both true. As these formulas are *subcontrariae*, '$\neg(p \wedge {\sim}p)$' is not CLuN-valid (cf. example 2).

Example 2: a synthetic tableau for '$\neg(p \wedge {\sim}p)$'

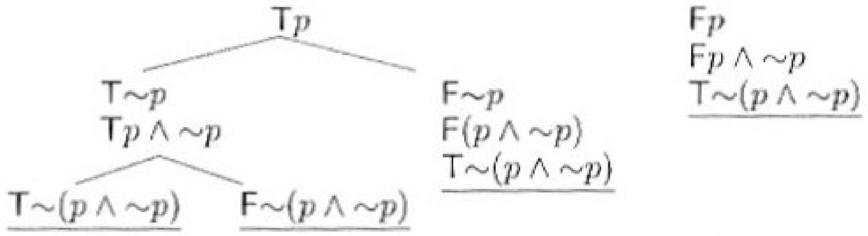


The formulas '$\neg{\sim}p$' and '$\neg p$' are *contrariae* and cannot be both true. Thus the formula '$\neg(\neg{\sim}p \wedge \neg p)$' is CLuN-valid (cf. example 3).

Example 3: a synthetic tableau for '$\neg(\neg{\sim}p \wedge \neg p)$'

$$\begin{array}{ll} \mathsf{T}p & \mathsf{F}p \\ \mathsf{F}\neg p & \mathsf{T}{\sim}p \\ \mathsf{F}(\neg{\sim}p \wedge {\sim}p) & \mathsf{F}\neg{\sim}p \\ \underline{\mathsf{T}\neg(\neg{\sim}p \wedge {\sim}p)} & \mathsf{F}(\neg{\sim}p \wedge {\sim}p) \\ & \underline{\mathsf{T}\neg(\neg{\sim}p \wedge {\sim}p)} \end{array}$$

Example 4: a synthetic tableau for '${\sim}(p \wedge {\sim}p)$'



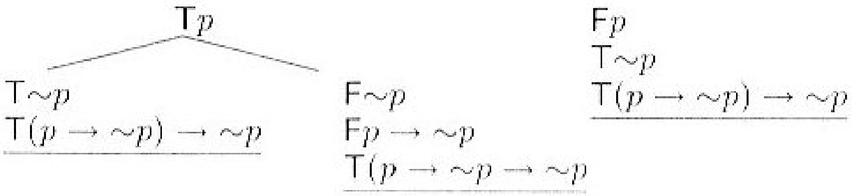
---

[7] Synthetic tableaux are defined as sets of sequences of wffs. Nevertheless, as in the examples here, it is convenient to represent them in a tree-like form, where every branch of a tree represents a certain synthetic inference of the tableau in question. The last formula of a synthetic inference is indicated by underlining.

On the other hand, law of excluded middle holds for *subcontrariae* formulas and its paraconsistent version (that is, the formula '$p \vee \sim p$') is CLuN-valid (cf. example 5).

Example 5: a synthetic tableau for '$p \vee \sim p$'

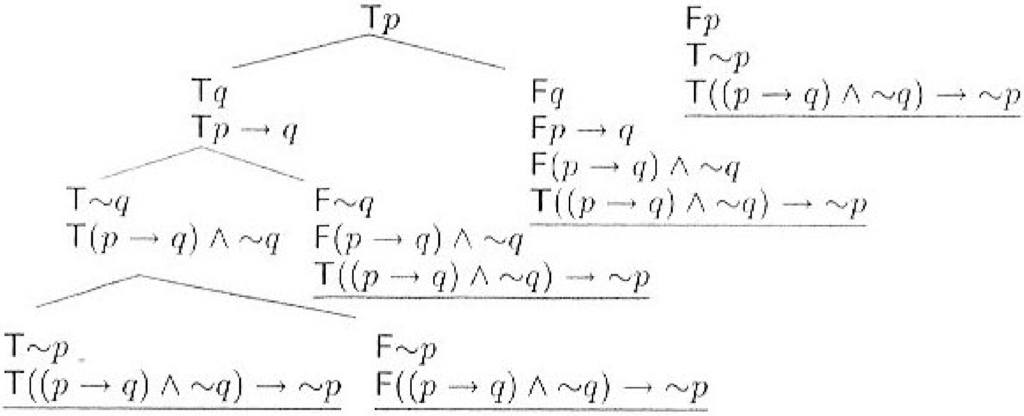
Example 6: a synthetic tableau for '$(p \rightarrow \sim p) \rightarrow \sim p$'



In the axiomatic setting, the formula '$p \vee \sim p$' (or, alternatively, the formula '$(p \rightarrow \sim p) \rightarrow \sim p$') is the only axiom that should be added to full positive classical logic in order to obtain CLuN.

Our final example is Modus Tollendo Tollens. This is one of the interesting features of CLuN: it invalidates MTT, while Modus Ponendo Ponens is CLuN-valid.

Example 7: a synthetic tableau for '$((p \rightarrow q) \wedge \sim q) \rightarrow \sim p$'



6. *Further applications*

Finally, let us shortly address the problem of further applications of STM in the CLuN-related contexts. There are two obvious possibilities. First one is that STM can be applied to other non-classical logics, that are related in a sense to CLuN, as, e.g. the logic ClaN, which allows for gaps with respect to negation[8], or the logic CluNs (cf. Batens (2003)). Second one is that STM can be applied as a direct method for some adaptive logics, especially for inconsistency-adaptive logics.

Section of Logic and Cognitive Science
Institute of Psychology
Adam Mickiewicz University in Poznań
Szamarzewskiego 89
60-589 Poznań, Poland
E-mail: `Mariusz.Urbanski@ifil.uz.zgora.pl`
`murbansk@wp.pl`

REFERENCES

Batens, D. (1986), Dialectical dynamics within formal logic, *Logique et Analyse*, 114, pp. 161–173.

Batens, D. (1989), Dynamic dialectical logics, in G. Priest, R. Routley, J. Norman (eds.), *Paraconsistent Logic. Essays on the Inconsistent*, Philosophia Verlag, Munich, pp. 187–217.

Batens, D. (1996), Functioning and teachings of adaptive logics, in J. van Benthem, F.H. van Eemeren, R. Grootendorst en F. Veltman (eds.), *Logic and Argumentation*, North Holland Publ., Amsterdam, pp. 241–254.

Batens, D. (1998), Inconsistency-adaptive logics, in: E. Orłowska (ed.), *Logic at Work. Essays Dedicated to the Memory of Helena Rasiowa*, Springer Verlag, Heidelberg/New York, pp. 445–472.

Batens, D. (2001), A General Characterization of Adaptive Logics, *Logique et Analyse*, 173–174–175, pp. 45–68.

Batens, D. (2003), Criteria Causing Inconsistencies. General Gluts as Opposed to Negation Gluts, *Logic and Logical Philosophy* 11/12, pp. 5–37.

Bocheński, I.M. (1951), *Ancient Formal Logic*, North Holland Publ., Amsterdam.

Łukasiewicz, J. (1951), *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic*, Oxford UP.

---

[8] In Batens (1996) this logic is called *paracomplete*.

Urbański, M. (2001a), Remarks on Synthetic Tableaux for Classical Propositional Calculus, *Bulletin of the Section of Logic* 30, No. 4, pp. 194–204.

Urbański, M. (2001b), Synthetic Tableaux and Erotetic Search Scenarios: Extension and Extraction, *Logique et Analyse*, 173–174–175, pp. 69–91.

Urbański, M. (2002a), *Tabele syntetyczne a logika pytań* [*Synthetic Tableaux and the Logic of Questions*], Maria Curie-Skłodowska University Press, Lublin.

Urbański, M. (2002b), Synthetic Tableaux for Łukasiewicz's Calculus Ł3, *Logique et Analyse* 177–178, pp. 155–173.

Wiśniewski, A. (2001), Questions and Inferences, *Logique et Analyse*, 173–174–175, pp. 5–43.

Wiśniewski, A. (2003), Erotetic Search Scenarios, *Synthese*, vol. 134(3), pp. 389–427.