The Sound and Complete Gentzen Deduction System for the Modalized Łukasiewicz Three-Valued Logic
Cungen Cao and Yuefei Sui
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Abstract
A modalized Łukasiewicz three-valued propositional logic will be proposed in this paper which there are three modalities \([t]\); \([m]\); \([f]\) to represent the three values \(t\); \(m\); \(f\); respectively. And a Gentzen-typed deduction system will be given so that the the system is sound and complete with respect to the Łukasiewicz three-valued semantics \(L_3\), which are given in soundness theorem and completeness theorem.

Keywords: Łukasiewicz three-valued logic, Modality, Soundness, Completeness

1. Introduction
The three-valued logics are traditional and have been studied in variant ways ([4–7, 14]). There are the following three-valued logics:

- **Bochvar’s three-valued logic** ([3, 6]), which logical language contains the logical connectives: \(\lor, \land, \rightarrow, \leftrightarrow, \equiv, =\), and the following semantics:

  \[
  \begin{array}{c|ccc|ccc|ccc}
  p \land q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p \lor q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p \rightarrow q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  \end{array}
  \]

- **Kleene’s three-valued logic** ([8, 14]), which logical language contains the logical connectives: \(\neg, \lor, \land, \rightarrow, \leftrightarrow, \equiv, =\), and the following semantics:

  \[
  \begin{array}{c|ccc|ccc|ccc}
  p \leftrightarrow q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p \equiv q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p = q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  \end{array}
  \]

- **Kleene’s three-valued logic** ([8, 14]), which logical language contains the logical connectives: \(\neg, \lor, \land, \rightarrow, \), and the following semantics:

  \[
  \begin{array}{c|ccc|ccc|ccc}
  \neg p & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p \land q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p \lor q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  p \rightarrow q & 1 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & -1 \\
  \end{array}
  \]
Łukasiewicz’s three-valued logic (\([10, 11]\)), which logical language contains the logical connectives: \(\neg, M, L, I, \lor, \land, \rightarrow, \leftrightarrow\), and the following semantics:

| p | \(\neg p\) | Mp | Ip |
|---|---|---|---|
| t | f | t | f |
| m | m | t | f |
| f | t | f | f |

\[
\begin{array}{|c|c|c|c|c|}
\hline
p & p \land q & p \lor q & p \rightarrow q & p \leftrightarrow q \\
\hline
q & t & m & f & t & m & f & t & m & f \\
\hline
\hline
p & t & m & f & t & m & f & t & m & f \\
\hline
m & m & m & f & t & m & m & m & m & m \\
\hline
f & f & m & m & m & f & t & t & t & m \\
\hline
\end{array}
\]

Post’s three-valued logic (\([12, 13]\)), which logical language contains the logical connectives: \(\neg, \lor, \land, \rightarrow, \leftrightarrow\), and the following semantics:

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\neg p & p \land q & p \lor q & p \rightarrow q & p \leftrightarrow q \\
\hline
p & t & m & f & t & m & f & t & m & f \\
\hline
q & t & f & t & f & f & m & m & m & f & f \\
\hline
m & m & t & m & f & t & m & m & f & t & m \\
\hline
f & f & m & m & m & f & t & t & f & m & f \\
\hline
\end{array}
\]

We will give a sound and complete Gentzen deduction system for Łukasiewicz’s three-valued propositional logic, that is, for any sequent \(\Gamma \Rightarrow \Delta\),

- The soundness theorem: If \(\vdash_{L_3} \Gamma \Rightarrow \Delta\) then \(\models_{L_3} \Gamma \Rightarrow \Delta\).
- The completeness theorem: If \(\models_{L_3} \Gamma \Rightarrow \Delta\) then \(\Gamma \vdash_{L_3} \Delta\).

This paper is organized as follows: the next section defines the basic elements in Łukasiewicz’s three-valued logic: the logical language, syntax and semantics; Section 3 gives a deduction system for Łukasiewicz’s three-valued propositional logic and proves the soundness theorem; Section 4 proves the completeness theorem for Łukasiewicz’s three-valued propositional logic, and Section 5 concludes the whole paper.

Our notation is standard, and a reference is \([9]\).

## 2. The Modalized Łukasiewicz Three-Valued Propositional Logic

Let the logical language contain the following symbols:

- propositional variables: \(p_0, p_1, \ldots\);
- modalities: \([t], [m], [f]\);
- unary logical connectives: \(\neg, M, L, I\), and
- binary logical connectives: \(\land, \lor, \rightarrow, \leftrightarrow\).

Formulas:

\[
A ::= p \quad \text{(atomic)}
\]

\[
\begin{align*}
&|[t]A_1|[m]A_1|[f]A_1 \quad \text{(modalized)} \\
&\neg A_1| M A_1 | L A_1 | I A_1 \quad \text{(unary connective)} \\
&A_1 \land A_2 | A_1 \lor A_2 | A_1 \\
&\triangleleft A_2 | A_1 \equiv A_2 \quad \text{(binary connective)}.
\end{align*}
\]

Let \(v\) be a function from the propositional variables to \(L_3 = \{t, m, f\}\), \(\leq\).

Define

\[
v(A) = \begin{cases} 
  v(p) & \text{if } A = p \\
  g_\ast(v(A_1)) & \text{if } A = [\ast]A_1 \\
  f_\circ(v(A_1)) & \text{if } A = \circ A_1 \\
  h_\bullet(v(A_1), v(A_2)) & \text{if } A = A_1 \bullet A_2,
\end{cases}
\]

where \(\ast \in \{t, m, f\}\), \(\circ \in \{\neg, M, L, I\}\), \(\bullet \in \{\land, \lor, \rightarrow, \leftrightarrow\}\) and

\[
\begin{array}{c|ccc|c|ccc|c|c|c}
\hline
& g_t & g_m & g_f & f_\neg & f_M & f_L & f_1 \\
\hline
\hline
\hline
\hline
\end{array}
\]

\[
\begin{array}{c|ccc|c|ccc|c|c|c}
& t & f & m & t & f & m & t & f \\
\hline
m & m & t & f & m & t & f & t & f \\
\hline
f & t & f & f & t & f & f & f & f \\
\hline
\end{array}
\]

\[
\begin{array}{c|ccc|c|ccc|c|c|c}
\hline
& g_t & g_m & g_f & f_\neg & f_M & f_L & f_1 \\
\hline
\hline
\hline
\hline
\end{array}
\]

\[
\begin{array}{c|ccc|c|ccc|c|c|c}
& t & f & m & t & f & m & t & f \\
\hline
m & m & t & f & m & t & f & t & f \\
\hline
f & t & f & f & t & f & f & f & f \\
\hline
\end{array}
\]
The Gentzen deduction system contains the following axioms and deduction rules.

For unary logical connectives:

1. **Negation**
   - \( \Gamma, A \Rightarrow \Delta \) (if \( \Gamma \Rightarrow \neg A \Rightarrow \Delta \))
   - \( \Gamma, \neg A \Rightarrow \Delta \) (\( \neg A \) is a tautology)
   - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)

2. **Disjunction**
   - \( \Gamma, A \lor B \Rightarrow \Delta \) (if \( \Gamma, A \Rightarrow \Delta \) and \( \Gamma, B \Rightarrow \Delta \))
   - \( \Gamma, A \Rightarrow \Delta \lor B \) (\( \Delta \lor B \) is true)
   - \( \Gamma, B \Rightarrow \Delta \) (\( \Delta \) is true)

3. **Conjunction**
   - \( \Gamma, A \land B \Rightarrow \Delta \) (if \( \Gamma, A \Rightarrow \Delta \) and \( \Gamma, B \Rightarrow \Delta \))
   - \( \Gamma, A \Rightarrow \Delta \land B \) (\( \Delta \land B \) is true)
   - \( \Gamma, B \Rightarrow \Delta \) (\( \Delta \) is true)

4. **Implication**
   - \( \Gamma, A \Rightarrow B \Rightarrow \Delta \) (if \( \Gamma, A \Rightarrow \Delta \) and \( \Gamma, B \Rightarrow \Delta \))
   - \( \Gamma, A \Rightarrow B \Rightarrow \Delta \) (\( A \Rightarrow B \) is true)

5. **Equivalence**
   - \( \Gamma, A \equiv B \Rightarrow \Delta \) (if \( \Gamma, A \Rightarrow B \Rightarrow \Delta \) and \( \Gamma, B \Rightarrow A \Rightarrow \Delta \))
   - \( \Gamma, A \equiv B \Rightarrow \Delta \) (\( A \equiv B \) is true)

Here, for the simplicity, we miss the deduction rules of the right side, and the same for the following rules for the unary logical connectives.

- **The deduction rules for unary logical connectives:**
  - \( \Gamma, A \Rightarrow \Delta \) (\( \neg \neg \neg \) \( A \))
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( \neg A \) is a tautology)
  - \( \Gamma, \neg A \Rightarrow \Delta \) (\( \neg A \) is a tautology)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)

- **The deduction rules for modalities and unary logical connectives:**
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( \neg A \) is a tautology)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)

3. **The Gentzen Deduction System**

The Gentzen deduction system contains the following axioms and deduction rules.

- **Axioms:**
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (if \( \Gamma, A \Rightarrow \Delta \) and \( \Gamma, \neg A \Rightarrow \Delta \))
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)

- **The deduction rules for modalities:**
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( \neg A \) is a tautology)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)
  - \( \Gamma, [\neg A] \Rightarrow \Delta \) (\( [\neg A] \) denotes \( \neg A \) in the two-valued propositional logic)

Given two sets \( \Gamma, \Delta \) of formulas, define

\[ v(\Gamma) = \min \{ v(A) : A \in \Gamma \} \]
\[ v(\Delta) = \max \{ v(A) : A \in \Delta \} \]

where the \( \leq \)-relation is that of \( L_3 \).

Given a sequent \( \Gamma \Rightarrow \Delta \), we say that \( v \) satisfies \( \delta \), denoted by \( v \vdash \Gamma \Rightarrow \Delta \) if \( v(\Gamma) \leq v(\Delta) \).

A sequent \( \delta \) is valid, denoted by \( \Gamma \vdash \Delta \), if for any assignment \( v \), \( v \vdash \Gamma \Rightarrow \Delta \).

Assume that \( [\neg A] \Rightarrow \Delta \) corresponds to the three values: +1, 0, -1, respectively. Then, we have

\[ [\neg A] \Rightarrow \Delta \]
\[ [\neg A] \Rightarrow \Delta \]
\[ [\neg A] \Rightarrow \Delta \]

where \( \Delta \) denotes \( \neg A \) in the two-valued propositional logic, respectively, and \( \neg A \) denotes \( \neg A \). Similarly we have the equivalences for \( \rightarrow \), \( \rightarrow \), \( = \) and \( \neq \).
\[ \Gamma, [m]A \Rightarrow \Delta (\neg [f]L) \]
\[ \Gamma, [\neg f]A \Rightarrow \Delta \]
\[ \Gamma, [m]A \Rightarrow \Delta (\neg [f]L) \]
\[ \Gamma, M[m]A \Rightarrow \Delta \]
\[ \Gamma, [t]A \vee [f]A \Rightarrow \Delta (M[t]L) \]
\[ \Gamma, [t]A \Rightarrow \Delta \]
\[ \Gamma, L[m]A \Rightarrow \Delta (L[m]L) \]
\[ \Gamma, [f]A \Rightarrow \Delta \]
\[ \Gamma, [I]A \Rightarrow \Delta (I[L]) \]
\[ \Gamma, [I]A \Rightarrow \Delta \]
\[ \Gamma, [I]A \Rightarrow \Delta (I[L]) \]
\[ \Gamma, [I]A \Rightarrow \Delta \]
\[ \Gamma, [I]A \Rightarrow \Delta (I[L]) \]

- The deduction rules for binary logical connectives:

\[ \Gamma, [A, B] \Rightarrow \Delta \]
\[ \Gamma, [A \land B] \Rightarrow \Delta \]
\[ \Gamma, [A \lor B] \Rightarrow \Delta \]
\[ \Gamma, [A \Rightarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Leftarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Rightarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Leftarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Rightarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Leftarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Rightarrow B] \Rightarrow \Delta \]
\[ \Gamma, [A \Leftarrow B] \Rightarrow \Delta \]

- The deduction rules for modalities and binary logical connectives:

\[ \diamond [t] + \diamond: \text{the same as the ones for logical connectives.} \]
\[ \diamond [m] + \diamond: \]
\[ \Gamma, [A, m] A \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]
\[ \Gamma, [m]A, [m]B \Rightarrow \Delta \]

- The deduction rules for unary and binary logical connectives:

\[ \neg \diamond + \bullet: \text{same as } \diamond + \diamond. \]
\[ \Gamma \Rightarrow \{ t \} A \wedge [m]B, [m]A \wedge [t]B, [m]A \wedge [x]B, \]
\[ \frac{\{ x \} A \wedge [m]B, \Delta}{\Gamma \Rightarrow \mathbb{I}(A \equiv B), \Delta} \quad (I \equiv R) \]

- The deduction rules for \( \Delta / \vee \).

\[ \frac{\Gamma, A \Rightarrow \Delta (L)_{1}}{\Gamma, A \wedge B \Rightarrow \Delta (R)} \]
\[ \frac{\Gamma, A \Rightarrow \Delta (L)_{2}}{\Gamma, A \Delta B \Rightarrow \Delta} \]
\[ \frac{\Gamma, A \Delta B \Rightarrow \Delta (L)_{2}}{\Gamma, A \Rightarrow \Delta (L)_{1}} \]
\[ \frac{\Gamma_{1}, A \Rightarrow \Delta, 1 \Gamma_{2} \Rightarrow B, \Delta_{2} (R)_{1}}{\Gamma_{1}, \Gamma_{2}, A \vee B \Rightarrow \Delta_{1}, \Delta_{2} (L)} \]
\[ \frac{\Gamma_{1}, \Gamma_{2}, A \vee B \Rightarrow \Delta_{1}, \Delta_{2} (L)}{\Gamma \Rightarrow \Delta} \]

**Definition 3.2.** \( \Gamma \vdash_{L_{3}} A \) if there is a sequence \( \Gamma_{1} \Rightarrow A_{1}, \ldots, \Gamma_{n} \Rightarrow A_{n} \) such that \( \Gamma_{n} \Rightarrow A_{n} = \Gamma \Rightarrow A \), and for each \( 1 \leq i \leq n, \Gamma_{i} \Rightarrow A_{i} \) is deduced from the previous sequents by one of the deduction rules.

**Theorem 3.3** (The soundness theorem). If \( \Gamma \vdash_{L_{3}} A \) then \( \models_{L_{3}} \Gamma \Rightarrow A \).

**Proof.** We prove that each axiom is valid and each deduction rule preserves the satisfiability.

To verify the validity of the axioms, assume that for any assignment \( \nu, \nu \models \Gamma, [m]p \). Then, \( \nu \models [m]p \), and so \( \nu \models [m]p, \Delta \). Similarly for other axioms.

To verify that \( (\models [m]l) \) preserves the validity, assume that for any assignment \( \nu, \nu \models \Gamma, [m]A \) implies \( \nu \models \Delta \). Because \( \nu \models [m]A \) implies \( \nu \models [m]A, \Delta \), for any assignment \( \nu, \nu \models [m]A, \Delta \). Similarly for cases of unary connectives.

To verify that \( (\models [m]_{L}) \) preserves the validity, assume that for any assignment \( \nu, \nu \models \Gamma_{1}, [t]A_{1}, [m]B, \lor \nu \models [t]A_{1}, [m]B \)
\[ \lor \nu \models [t]A_{1}, \lor \nu \models [m]B, \lor \nu \models [m]B \]. By the assumption, \( \nu \models \Gamma_{1}, [t]A_{1}, [m]B, \lor \nu \models [t]A_{1}, [m]B, \lor \nu \models [t]A_{1}, [m]B, \lor \nu \models [m]B, [m]A \wedge [t]B, [m]A \wedge [x]B, [m]A \wedge [x]B \).

**4. The completeness theorem**

**Theorem 4.1** (The completeness theorem). If \( \models_{L_{3}} \Gamma \Rightarrow A \)
then \( \Gamma \vdash_{L_{3}} A \).

**Proof.** Let \( \delta = \Gamma \Rightarrow A \). Define a tree, called the reduction tree for \( \delta \), denoted by \( T(\delta) \), from which we can obtain either a proof of \( \delta \) or a show of the nonvalidity of \( \delta \).

This reduction tree \( T(\delta) \) contains a sequent at each node, and is constructed in stages as follows.

- **Stage 0:** \( T_{0}(\delta) = \{ \delta \} \).
- **Stage** \( k(k > 0) : T_{k}(\delta) \) is defined by cases.
- **Case 0:** If \( \Gamma \Rightarrow \Delta \) is an axiom, write nothing above \( \Gamma \Rightarrow \Delta \).
- **Case 1:** Every topmost sequent \( \Gamma \Rightarrow \Delta \) in \( T_{k-1}(\delta) \) is an axiom. Then, stop.
- **Case 2:** Not Case 1. \( T_{k}(\delta) \) is defined as follows. Let \( \Gamma \Rightarrow \Delta \) be any topmost sequent of the tree which has been defined by stage \( k \). Then, write down

\[ \Gamma_{1}, [t]A_{1}, \ldots, [t]A_{n} \Rightarrow \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \( ([t][t]l) \) reduction has been applied to \( [t][t]A_{1}, \ldots, [t][t]A_{n} \).

**Subcase ([t][t]R).** Let \( [t][t]A_{1}, \ldots, [t][t]A_{n} \) be all the formulas in \( \Gamma \) whose outermost logical symbol is \( [t][t] \), and to which no reduction has been applied in previous stages. Then, write down

\[ \Gamma \Rightarrow [t]A_{1}, \ldots, [t]A_{n} \]
above \( \Gamma \Rightarrow \Delta \). We say that a \( ([t][t]R) \) reduction has been applied to \( [t][t]A_{1}, \ldots, [t][t]A_{n} \).

**Subcase (ML L).** Let \( ML A_{1}, \ldots, ML A_{n} \) be all the formulas in \( \Gamma \) whose outermost logical symbol is \( ML \), and to which no
reduction has been applied in previous stages. Then, write down
\[ \Gamma, [t]A_1, ..., [t]A_n \Rightarrow \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \((ML^L)\) reduction has been applied to \(MLA_1, ..., MLA_n\).

Subcase \((ML^R)\). Let \(MLA_1, ..., MLA_n\) be all the formulas in \(\Delta\) whose outermost logical symbol is \(ML\), and to which no reduction has been applied in previous stages. Then, write down
\[ \Gamma \Rightarrow [t]A_1, ..., [t]A_n, \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \((ML^R)\) reduction has been applied to \(MLA_1, ..., MLA_n\).

Subcase \((L[t]^L)\). Let \(L[t]A_1, ..., L[t]A_n\) be all the formulas in \(\Delta\) whose outermost logical symbol is \(L[t]\), and to which no reduction has been applied in previous stages. Then, write down
\[ \Gamma, [t]A_1, ..., [t]A_n \Rightarrow \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \((L[t]^L)\) reduction has been applied to \(L[t]A_1, ..., L[t]A_n\).

Subcase \((L[t]^R)\). Let \(L[t]A_1, ..., L[t]A_n\) be all the formulas in \(\Delta\) whose outermost logical symbol is \(L[t]\), and to which no reduction has been applied in previous stages. Then, write down
\[ \Gamma \Rightarrow [t]A_1, ..., [t]A_n, \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \((L[t]^R)\) reduction has been applied to \(L[t]A_1, ..., L[t]A_n\).

Similar for other cases of unary connectives.

Subcase \(([m] \land ^L)\). Let \([m](A_1 \land B_1), ..., [m](A_n \land B_n)\) be all the statements in \(\Gamma\) whose outermost logical symbol is \([m] \land\), and to which no reduction has been applied in previous stages by any \(([m] \land ^L)\). Then, for each partition \(\{I_1, I_2, I_3\}\) of \(\{1, ..., n\}\), write down
\[ \Gamma, [t]A_i : i \in I_1, [m]B_i : i \in I_1 \cup I_3, [m]A_i : i \in I_2 \cup I_3, [t]B_i : i \in I_2 \Rightarrow \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \(([m] \land ^L)\) reduction has been applied to \([m](A_1 \land B_1), ..., [m](A_n \land B_n)\).

Subcase \(([m] \land ^R)\). Let \([m](A_1 \land B_1), ..., [m](A_n \land B_n)\) be all the statements in \(\Delta\) whose outermost logical symbol is \([m] \land\), and to which no reduction has been applied in previous stages by any \(([m] \land ^R)\). Then, write down
\[ \Gamma \Rightarrow [t]A_1 \land [m]B_1, [t]A_1 \land [t]B_1, [m]A_1 \land [m]B_1, ..., [t]A_n \land [m]B_n, [m]A_n \land [t]B_n, [m]A_n \land [m]B_n, \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \(([m] \land ^R)\) reduction has been applied to \([m](A_1 \land B_1), ..., [m](A_n \land B_n)\).

Subcase \(([m] \lor ^L)\). Let \([m](A_1 \lor B_1), ..., [m](A_n \lor B_n)\) be all the statements in \(\Gamma\) whose outermost logical symbol is \([m] \lor\), and to which no reduction has been applied in previous stages by any \(([m] \lor ^L)\). Then, for each partition \(\{I_1, I_2, I_3\}\) of \(\{1, ..., n\}\), write down
\[ \Gamma, [m]A_i : i \in I_1 \cup I_2, [m]B_i : i \in I_1 \cup I_3, [m]A_i : i \in I_2 \cup I_3, [t]B_i : i \in I_2 \Rightarrow \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \(([m] \lor ^L)\) reduction has been applied to \([m](A_1 \lor B_1), ..., [m](A_n \lor B_n)\).

Subcase \(([m] \lor ^R)\). Let \([m](A_1 \lor B_1), ..., [m](A_n \lor B_n)\) be all the statements in \(\Delta\) whose outermost logical symbol is \([m] \lor\), and to which no reduction has been applied in previous stages by any \(([m] \lor ^R)\). Then, write down
\[ \Gamma \Rightarrow [m]A_1 \lor [m]B_1, [m]A_1 \lor [t]B_1, [m]A_1 \lor [m]B_1, ..., [m]A_n \lor [m]B_n, [m]A_n \lor [t]B_n, \Delta \]
above \( \Gamma \Rightarrow \Delta \). We say that a \(([m] \lor ^R)\) reduction has been applied to \([m](A_1 \lor B_1), ..., [m](A_n \lor B_n)\).
Subcase (M^L). Let \( M(A_1 \lor B_1), \ldots, M(A_n \lor B_n) \) be all the statements in \( \Delta \) whose outermost logical symbol is \( M \lor \), and to which no reduction has been applied in previous stages by any (M^L). Then, for each partition \( \{I_1, \ldots, I_8\} \) of \( \{1, \ldots, n\} \), write down

\[
\Gamma, [t]A_i: i \in I_1 \cup I_2 \cup I_3, [t]B_i: i \in I_4 \cup I_7, [m]A_i: i \in I_1 \cup I_5 \cup I_6, [m]B_i: i \in I_2 \cup I_4 \cup I_8, [f]A_i: i \in I_7 \cup I_8, [f]B_i: i \in I_3 \cup I_6 \Rightarrow \Delta
\]

above \( \Gamma \Rightarrow \Delta \). We say that a (M^L) reduction has been applied to \( M(A_1 \lor B_1), \ldots, M(A_n \lor B_n) \).

Subcase (M^R). Let \( M(A_1 \lor B_1), \ldots, M(A_n \lor B_n) \) be all the statements in \( \Delta \) whose outermost logical symbol is \( M \lor \), and to which no reduction has been applied in previous stages by any (M^R). Then, write down

\[
\Gamma \Rightarrow \Delta \quad \text{and} \quad \Delta \Rightarrow \Delta \quad \text{above} \quad \Gamma \Rightarrow \Delta \quad \text{and} \quad \Delta \Rightarrow \Delta
\]

above \( \Gamma \Rightarrow \Delta \). We say that a (M^R) reduction has been applied to \( M(A_1 \lor B_1), \ldots, M(A_n \lor B_n) \).

Subcase (M^L). Let \( M(A_1 \lor B_1), \ldots, M(A_n \lor B_n) \) be all the statements in \( \Delta \) whose outermost logical symbol is \( M \lor \), and to which no reduction has been applied in previous stages by any (M^L). Then, write down

\[
\Gamma, A_1, B_1, \ldots, A_n, B_n \Rightarrow \Delta
\]

above \( \Gamma \Rightarrow \Delta \). We say that a (M^L) reduction has been applied to \( (A_1 \lor B_1), \ldots, (A_n \lor B_n) \).

Subcase (M^R). Let \( M(A_1 \lor B_1), \ldots, M(A_n \lor B_n) \) be all the statements in \( \Delta \) whose outermost logical symbol is \( M \lor \), and to which no reduction has been applied in previous stages by any (M^R). Then, write down

\[
\Gamma \Rightarrow C_1, \ldots, C_n, \Delta
\]
segment of $\sigma$ such that $\beta = \Gamma', [m](A_1 \land A_2) \Rightarrow \Delta'$ for some $\Gamma'$ and $\Delta'$. Then, there is a segment $\gamma$ of $\sigma$ such that $\beta$ is a segment of $\gamma$ and $\gamma$ is one of the following forms:

$$
\Gamma', [t] A_1, [m] A_2 \Rightarrow \Delta',
$$

$$
\Gamma', [m] A_1, [t] A_2 \Rightarrow \Delta',
$$

$$
[\Gamma, [m] A_1, [m] A_2 \Rightarrow \Delta'],
$$

say $\gamma = \Gamma', [m] A_1, [m] A_2 \Rightarrow \Delta'$. By induction assumption, $v \models \Gamma', [m] A_1, [m] A_2$ and $v \not\models \Delta'$. Then, by the definition of satisfaction, $v \models \Gamma', [m](A_1 \land A_2)$ and $v \not\models \Delta'$.

Case $A = [m](A_1 \land A_2) \in \Delta$. Let $\beta$ be the least-length segment of $\sigma$ such that $\beta = \Gamma' \Rightarrow [m](A_1 \land A_2)$, $\Delta'$. Then, there are segments $\gamma_1, \gamma_2, \gamma_3$ of $\sigma$ such that $\beta$ is a segment of $\gamma_1, \gamma_2, \gamma_3$ and $\gamma_1 = \Gamma' \Rightarrow [t] A \Delta [m] B, \Delta'; \gamma_2 = \Gamma' \Rightarrow [m] A \Delta [t] B, \Delta'$ and $\gamma_3 = \Gamma' \Rightarrow [m] A \Delta [m] B, \Delta'$. By the induction assumption, $v \models \Gamma'$ and $v \not\models [t] A \Delta [m] B, \Delta'; v \not\models [m] A \Delta [t] B, \Delta'; v \not\models [m] A \Delta [m] B, \Delta'$, i.e., $v \not\models [m](A_1 \land A_2), \Delta'$.

Similarly for other cases.

This completes the proof. \qed

5. Conclusions

In this paper we gave a modalized Łukasiewicz three-valued propositional logic and a Gentzen deduction system was constructed such that the soundness theorem and the completeness theorem hold in Łukasiewicz three-valued semantics of the modalized propositional logic.

In practical applications, we use the traditional fuzzy logic in which only two truth-values $t$ and $f$ are considered in the deduction, even though in semantics, a formula can have any values as the truth-values. In the Gentzen deduction system given in this paper, each truth-value contributes to the deduction, which makes the system a little clumsy. As a system which can be implemented in computer, we hope the Gentzen deduction system is used in practice in near future.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This work was supported by the National Science Foundation of China (under grant Nos. 91224006 and 61173063) and the Ministry of Science and Technology (under grant No.201303107).

References

[1] A. Avron, “Natural 3-valued logics: characterization and proof theory,” Journal of Symbolic Logic, vol. 56, no. 1, pp. 276-294, 1991. [http://dx.doi.org/10.2307/2274919]

[2] A. Avron, “Gentzen-type systems, resolution and tableaux,” Journal of Automated Reasoning, vol. 10, no. 2, pp. 265-281,1993. [http://dx.doi.org/10.1007/BF00881838]

[3] D. A. Bochvar and M. Bergmann, “On a three-valued logical calculus and its application to the analysis of the paradoxes of the classically extended functional calculus,” History and Philosophy of Logic, vol. 2, no. 1-2, pp. 87-112, 1981. [http://dx.doi.org/10.1080/01445348108837023]

[4] M. Fitting, “Many-valued modal logics II,” Fundamenta Informaticae, vol. 17, no. 1-2, pp. 55-73, 1992.

[5] S. Gottwald, A Treatise on Many-Valued Logics. Baldock: ResearchStudies Press, 2001.

[6] S. Gottwald, “Many-Valued Logic,” Available [http://plato.stanford.edu/entries/logic-manyvalued/]

[7] R.Hahnle, “Advanced many-valued logics,” in Handbook of Philosophical Logic, D. M. Gabbay and F. Guenthner, Eds. Dordrecht: Springer, 2001, pp. 297-395. [http://dx.doi.org/10.1007/978-94-017-0452-6_5]

[8] S. C. Kleene, “On notation for ordinal numbers,” Journal of Symbolic Logic, vol. 3, no. 4, pp. 150-155,1938. [http://dx.doi.org/10.2307/2267778]

[9] W. Li, Mathematical Logic: Foundations for Information-Science. Basel: BirkhauserVerlag AG, 2010.

[10] J. Łukasiewicz, “O logice trojwartosciowej [On three-valued logic],” Ruch filozoficzny, vol. 5, pp. 170-171, 1920.

[11] J. Łukasiewicz, “Selected Works,” in Studies in Logic and the Foundations of Mathematics.L. Borkowski, Ed. Amsterdam:North-Holland and Warsaw, 1970.

[12] E. L. Post, “Determination of all closed systems of truthtables,” Bulletin American Mathematical Society, vol. 26, pp. 437, 1920.

[13] E. L. Post, “Introduction to a general theory of elementary propositions,” American Journal of Mathematics, vol. 43, no. 3, pp. 163-185, 1921. [http://dx.doi.org/10.2307/2370324]
[14] A. Urquhart,”Basic many-valued logic,” in *Handbook of philosophical logic (vol. 2)*, D. M. Gabbay and F. Guenthner, Eds. Dordrecht: Springer, 2001, pp. 249-295. 
http://dx.doi.org/10.1007/978-94-017-0452-6_4

[15] W. Zhu, “Independence issues of the propositional connectives in medium logic systems MP and MP,” in *A Friendly Collection of Mathematical Papers I*. Changchun: Jilin University Press, 1990.

[16] J. Zhu, X. Xiao, and W. Zhu, “A survey of the development of medium logic calculus system and the research of its semantics,” in *A Friendly Collection of Mathematical Papers I*. Changchun: Jilin University Press, 1990.

[17] W. Zhu and X. Xiao, *An Introduction to Foundations of Mathematics*. Nanjing: Nanjing University Press, 1996.

**Cungen Cao** received his M.S. and Ph.D. degree in software from the Institute of Mathematics, Chinese Academy of Sciences in 1989 and 1993 respectively. He is a full-time professor of ICT, CAS, since 2000. He is leading a research group of large-scale knowledge engineering and knowledge-intensive applications.

**Yuefei Sui** is a Professor in the Chinese Academy of Sciences (ICT), Chinese Academy of Sciences. His main interests include knowledge representation, applied logic and the theory of computability.