Refutation of the Diproche system for mistake diagnosis in the didactics of mathematics

© Copyright 2020 by Colin James III  All rights reserved.

Abstract: We evaluate 12 logical fallacies flagged by prover Diproche with four as incorrect, refuting it as bivalent, with its vector space results forming a *non* tautologous fragment of the universal logic \( VŁ4 \).

We assume the method and apparatus of Meth8/VŁ4 with \( \top \) as tautology as the designated proof value, \( \bot \) as contradiction, \( \perp \) as truthity (non-contingency), and \( \perp \) as falsity (contingency). The 16-valued truth table is row-major and horizontal, or repeating fragments of 128-tables, sometimes with table counts, for more variables. (See ersatz-systems.com.)

LET \(~ \) Not, \( \neg \); \(+ \) Or, \( V \), \( U \), \( \cup \); \(- \) Not Or; \( \& \) And, \( \wedge \), \( \cap \), \( \circ \); \( \setminus \) Not And;
> Imply, greater than, \( \rightarrow \), \( \Rightarrow \), \( \supset \), \( \supseteq \); \(< \) Not Imply, less than, \( \in \), \( \subset \), \( \not\in \), \( \not\subset \), \( \subseteq \); \(= \) Equivalent, \( \equiv \), \( \cong \), \( \simeq \); \(\equiv \) Not Equivalent, \( \neq \), \( \oplus \);
% possibility, for one or some, \( \exists \), \( \exists! \), \( \Diamond \), \( \Box \); \# necessity, for every or all, \( \forall \), \( \square \), \( \lozenge \);
\( (z=z) \) \( \top \) as tautology, \( \top \), ordinal 3; \( (z\not=\not z) \) \( \bot \) as contradiction, \( \emptyset \), Null, \( \bot \), zero;
\( (\%z>\#z) \) \( \perp \) as non-contingency, \( \Delta \), ordinal 1; \( (\%z<\#z) \) \( \perp \) as contingency, \( \forall \), ordinal 2;
\( \sim \) \( (y < x) \) \( (x \leq y) \), \( (x \subseteq y) \), \( (x \subseteq y) \); \( (A=B) \) \( (A\sim B) \).
Note for clarity, we usually distribute quantifiers onto each designated variable.


Abstract. The Diproche system, an automated proof checker for natural language proofs specifically adapted to the context of exercises for beginner’s students similar to the Naproche system by … others, uses a modification of an automated theorem prover which uses common formal fallacies instead of sound deduction rules for mistake diagnosis. We briefly describe the concept of such an ‘Anti-ATP’ and explain the basic techniques used in its implementation.

3 Automatization of mistake diagnosis
3.1 Diagnosis of logical fallacies: the “Anti-ATP”
LET $p, q, r$: $A$ or $\varphi$, $B$ or $x$, $C$ or $y$.

$(\neg p \land (p < q)) > \neg q$ ; $TTTT$ $TTTT$ $TTTT$ $TTTT$ $(3.1.1.2.2)$

$((\# q \land \% r) \land p) > ((\% r \land \# q) \land p)$; $TTTT$ $TTTT$ $TTTT$ $TTTT$ $(3.1.6.2)$

$(p < q) > (q < p)$ ; $TTTT$ $TTTT$ $TTTT$ $TTTT$ $(3.1.8.2.2.2)$

$((p < q) \land (q < r)) > (p < r)$ ; $TTTT$ $TTTT$ $TTTT$ $TTTT$ $(3.1.9.2)$

**Remark 3.1**: Eq. 3.1.1.2.2 for inverse contraposition *not* contrary. Eq. 3.1.6.2 for quantifier switch is *not* contrary because the rule does not know that respective quantified and modal operators are equivalent (and distributive). Eq. 3.1.8.2.2.2 is *not* contrary because the rule does not know subset and element relations are equivalent. Eq. 3.1.9.2 is *not* contrary because the rule does not know that element relation is transitive.