Refutation of replacing classical Logic with free logic

Abstract: We evaluated 12 equations for the assertions with none tautologous. Therefore this conjecture is a *non* tautologous fragment of the universal logic VŁ4.

We assume the method and apparatus of Meth8/VŁ4 with Tautology as the designated proof value, **F** as contradiction, N as truthity (non-contingency), and C 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, ¬ ; + Or, ∨, ∪, ⊔ ; - Not Or; & And, ∧, ∩, ⊓, ·; \ Not And;
> Imply, greater than, →, ⇒, ↦, ≻, ⊃, ↠ ; < Not Imply, less than, ∈, ≺, ⊂, ⊬, ⊭, ↞, ≲ ;
= Equivalent, ≡, :=, ⇔, ↔, ≜, ≈, ≃ ; @ Not Equivalent, ≠;
% possibility, for one or some, ∃, ◊, M; # necessity, for every or all, ∀, □, L;
(z=z) T as tautology, ⊤, ordinal 3; (z@z) **F** as contradiction, Ø, Null, ⊥, zero;
(%z>#z) N as non-contingency, Δ, ordinal 1; (%z<#z) C as contingency, ∇, ordinal 2;
~(y < x) (x ≤ y), (x ⊆ y); (A=B) (A~B).
Note for clarity, we usually distribute quantifiers onto each designated variable.

From: Bornali, P. (2019). Proposal of replacing classical logic with free logic for reasoning with non-referring names in ordinary discourse. vixra.org/pdf/1905.0358v1.pdf [no email published]

Abstract: Reasoning carried out in ordinary language, can not avoid using non-referring names if occasion arises. Semantics of classical logic does not fit well for dealing with sentences with non-referring names of the language. The principle of bivalence does not allow any third truth-value, it does not allow truth-value gap also. The outcome is an ad hoc stipulation that no names should be referentless. The aim of this paper is to evaluate how far free logic with supervaluational semantics is appropriate for dealing with the problems of non-referring names used in sentences of ordinary language, at the cost of validity of some of the classical logical theses/ principles.

3.1. Presupposition as a semantic relation … is as follows: If A and B are two propositions, then a characterization of presupposition can be given in a language as,

A presupposes B iff A is neither true nor false unless B is true. (3.1.1.1)

LET p, q: A, B. (This makes for shorter table results for propositions and not theorems.)

((q=(q=q))>~(~((p=~(p=p))+(p=~(p@p)))=(p=p)))>(p>q) ;
T**F**TT T**F**TT T**F**TT T**F**TT (3.1.1.2)

This is equivalent to, If A is true, then B is true and, If A is false, then B is true ... (3.1.2.1)

((p=(p=p))>(q=(q=q)))&((p=(p@p))>(q=(q=q))) ;
FFTT **FF**TT **FF**TT **FF**TT (3.1.2.2)

Remark 3.1.1.2-3.12.2: Eqs. 3.1.1.2 and 3.1.2.2 are *not* tautologous and *not* equivalent, as asserted, hence refuting those two conjectures.

Presupposition is different from other semantic relations, e.g., implication and necessitation.

Implication is defined as the logical truth of

'A⊃B'(~A∨B). (3.1.3.1)

(p>q)&(~p+q) ; T**F**TT T**F**TT T**F**TT T**F**TT (3.1.3.2)

For implication *modus tollens* is accepted as valid, whereas in case of presupposition it doesn't hold, since the analogue of *modus tollens* with respect to presupposition:

A presupposes B (not B) Therefore, (not A) is not valid; if both the premises are true, the conclusion is not true (i.e. neither true nor false). (3.1.4.1)

((p>q)>~q)>~p ; T**F**T**F** T**F**T**F** T**F**T**F** T**F**T**F** (3.1.4.2)

Another distinction is that the argument:

A presupposes B (not A) Therefore, B is valid in case of presupposition, since if the premises are true, so is the conclusion; whereas, for implication this argument doesn't hold. (3.1.5.1)

((p>q)>~p)>q ; **FF**TT **FF**TT **FF**TT **FF**TT (3.1.5.2)

However, presupposition and implication have something in common, which is, if A either resupposes or implies B then the argument from A to B is valid. (3.1.6.1)

Remark 3.1.6.1: We map resupposes as either Eqs. 3.1.4.1 or 3.1.5.1, or 3.1.3.1, as valid.

((((p>q)>~q)>~p)+(((p>q)>~p)>q))+((p>q)&(~p+q)) ;
T**F**TT T**F**TT T**F**TT T**F**TT (3.1.6.2)

3.3. Shortcomings of supervaluation semantics:
Considering the above case, where 'a' is denoting and 'b' is not,

'(∀x)Px⊃Pa' is true, (3.3.1.1)

LET p, q, r, s: p, x, a, b.

(r>(p=p))>((p&#q)>(p&r)) ; TTTT TTTC TTTT TTTC (3.3.1.2)

though '(∀x)Px⊃Pa' is not. (3.3.2.1)

(s>(p@p))>((p&#q)>(p&s)) ; TTTC TTTC TTTT TTTT (3.3.2.2)

However, in standard first order predicate logic (FOP) both are true as endorsed by UI rule, known as the principle of Specification. This is however quite expected in a system of free logic.

Remark 3.3: Eqs. 3.3.1.2 and 3.3.2.2 as rendered are *not* tautologous and *not* contradictory, thereby refuting four conjectures in FOP: two as true and false, and two as true.

All conjectures evaluated in 12 eqs. are *not* tautologous, and refutes replacing classical logic with free logic.