Refutation of the logical syntax of IT architectures

© Copyright 2020 by Colin James III   All rights reserved.

Abstract: We evaluate two conditions in the antecedent of a definition, both separately and as respective negations, then proceed to a definition of the consequent. The conjecture to define the symmetric binary relations of an n-tier hierarchy is not tautologous. The conjecture with substitutions in a subsequent lemma is also denied. This refutes the conjecture of the logical syntax of IT architecture. These results form a non tautologous fragment of the universal logic \( \mathbb{V}_4 \).

We assume the method and apparatus of Meth8/\( \mathbb{V}_4 \) with Tautology as the designated proof value, \( \top \) 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 us directly start with the definition of an architecture. Our notation follows standard model theory ..

\[
\begin{align*}
\text{Definition 9} \ (\text{elementary } n\text{-tier architecture}). & \text{ Let } T_n = \{1, 2, ..., n\} \text{ and define the symmetric binary relation } T^L \ (\text{"linked to"}) \text{ over } T^2_n \text{ as follows:} \\
& |i - j| \leq 1 \Rightarrow T^L(i, j), \\
& |i - j| > 1 \Rightarrow \neg T^L(i, j)
\end{align*}
\]

Then the B\&C architecture \( J_n = \langle T_n, \{T^L\} \rangle \) is called the elementary \( n\)-tier architecture. \( (2.9.1.1), (2.9.2.1) \)

Let \( p, q, r: i, j, T^i(i, j) \). \( (2.9.1.2) \)

\[
\begin{align*}
\sim((%s>\#s)>(p-q))>r ; & \quad \text{FNNN TTTT FNNN TTTT} \\
((p-q)>(%s>\#s))>\sim r ; & \quad \text{TTTT CFFF TTTT CFFF}
\end{align*}
\]

Remark 2.9.1.2, .2.2: Eqs. 2.9.1.2 and 2.9.2.2 as rendered are not tautologous, hence refuting the definition of the symmetric binary relation and subsequent conjectures. If the definitions are respective negations, then the conjunction should be contrary. \( (2.9.3.1) \)

\[
\begin{align*}
\sim((%s>\#s)>(p-q))>r) \& (((p-q)>(%s>\#s))>\sim r) ; & \quad \text{FNNN CFFF FNNN CFFF} \quad (2.9.3.2)
\end{align*}
\]
**Remark 2.9.3.2:** Eq. 2.9.3.2 is *not* contrary, hence refuting the antecedent definitions as respective negations.

We attempt to resuscitate Def. 9 by completing the argument according to the apparent intention. The comma in the text separating Eqs. 2.9.1.1 and 2.9.2.1 taken to mean "And" should read "Or" as 2.9.1.1 or 2.9.2.1.  

\[(2.9.4.1)\]

\[
\neg((\%s>\#s)<(p-q))>r)+(((p-q)>(\%s>\#s))>\neg r) ; \\
TTTT TTTT TTTT TTTT 
\]

\[(2.9.4.2)\]

The goal was apparently to express the antecedent in Def. 9 as a tautology.

The argument in Def. 9 then becomes the antecedent as true to imply the consequent of B&C Architecture (B&C) such that \(T\) implies (B&C). For this to hold, the goal is for \(T\) to imply \(T\) and not for the disallowed \(T\) to imply \(F\).

B&C is defined in Def. 5 below:

**Definition 5** (boxes & connectors (B&C) architecture). An architecture \(\mathcal{A} = (A, R \subseteq \mathcal{P}(A^2), \emptyset)\) is called a boxes & connectors (B&C) architecture.

A B&C architecture is called **connected** if \(\forall a \in A : \exists R \in R, \exists a^* \in A : R(a,a^*) \vee R(a^*,a) \wedge a \neq a^*.\) Otherwise it is called **disconnected**.

We shall often write \(\mathcal{A} = (A, R)\) for a B&C architecture in the following.

Architects frequently used so-called \(n\)-tier architectures which can be rigorously defined as followed.

\[(2.5.1)\]

\[\text{LET t, u, v, w, x: a, a^*, A, R, R.}\]

\[\text{((#t<v>)(%w<x>&(%u<v))>((w&(t&u)) +((w&(t&u))&(t@u)))) ;}\]

\[
\text{FFFF FFFF FFFF FFFF ( 1 }) 2 \} 8 \\
\text{NNNN NNNN NNNN NNNN ( 1 }) \}
\]

\[
\text{FFFF FFFF FFFF FFFF ( 5 }) 1 \} \\
\text{NNNN NNNN NNNN NNNN ( 1 }) \}
\]

\[
\text{FFFF FFFF FFFF FFFF ( 1 }) \}
\]

\[
\text{TTTT TTTT TTTT TTTT ( 1 }) \}
\]

\[
\text{FFFF FFFF FFFF FFFF ( 3 }) \}
\]

\[
\text{TTTT TTTT TTTT TTTT ( 1 }) \}
\]

\[(2.5.2)\]

**Remark 2.5.2:** Eq 2.5.2 is *not* tautologous as required to confirm Def. 9. In fact, completing the argument for Def. 9 with the Eq. 2.9.4.1 as antecedent and 2.5.1 as consequent produces the same result as 2.5.2. This means Def. 9 is *not* tautologous using Def. 5 which is also *not* tautologous, hence refuting the claimed conjecture of a syntax for IT architecture. We include a subsequent Lemma 1 which by substitutions in its proof is similarly denied for the same reasons.

### 3.2 Theorems

Here we demonstrate that our theory of (IT) architectures also provides a mathematical framework for reasoning (and proving) facts about architectures.

...
Lemma 1. Every $n$-tier B&C architecture $\mathcal{A}$ is homomorphic to the elementary $n$-tier architecture $\mathcal{T}_n$, that is there exists a homomorphism $h : \mathcal{A} \rightarrow \mathcal{T}_n$.

Proof. Let $\mathcal{A} = (A, R)$ be a $n$-tier B&C architecture. Then, by definition, every element $a \in A$ belongs to exactly one tier $C_i$ with $1 \leq i \leq n$. We also have $R \subseteq A^2$ because of the ”boxes & connectors“ property. Then define the map $h : \mathcal{A} \rightarrow \mathcal{T}_n$ as follows:

$$h(\cdot) : \begin{cases} a \in C_i \quad \Rightarrow \quad i \in T_n, \\ R \in \mathbf{R} \quad \Rightarrow \quad T^L \end{cases}$$

Assume $R(c_i, c_j)$ with $c_i \in C_i$ and $c_j \in C_j$. Then $h(c_i) = i$ and $h(c_j) = j$. Because $\mathcal{A}$ is an $n$-tier architecture, we have

$$|i - j| \leq 1 \Rightarrow |h(c_i) - h(c_j)| \leq 1 = T^L(h(c_i), h(c_j)) = h(R)(h(c_i), h(c_j))$$

Therefore $h$ is a homomorphism. \qed