Refutation of the unified language for linear and relational algebra (Lara)

© Copyright 2019 by Colin James III All rights reserved.

Abstract: We evaluate two proofs of Th. 3 of the Lara language as not tautologous. By extension the Lara language as a unification of linear and relational algebra is also refuted. The conjectures form 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, ∃, ø, 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), ( x ⊂ y); (A=B) (A~B).
Note for clarity, we usually distribute quantifiers onto each designated variable.

pbarcelo@ing.puc.cl, nhiguera@dcc.uchile.cl, jperez@dcc.uchile.cl, bsuberca@dcc.uchile.cl

Abstract We study the expressive power of the Lara language – a recently proposed unified model for expressing relational and linear algebra operations – both in terms of traditional database query languages and some analytic tasks often performed in machine learning pipelines. We start by showing Lara to be expressive complete with respect to first-order logic with aggregation. ...

4 Expressive completeness of Lara with respect to FO_{Agg}

Theorem 3. For every expression \( e[K, V] \) of LARA(\( \Omega \)) there is a formula \( \phi_e(\bar{x}, \bar{i}) \) of \( \text{FO}_{\text{Agg}}(\Psi_\Omega) \) such that \( e^D = \phi_e^D \), for every LARA database \( D \).

Proof. By induction on \( e \). The full proof is in the appendix. The basis cases are simple. Instead of the inductive case we present an example of how the union operation is translated, as this provides a good illustration of the main ideas behind the proof.

Let us define a formula \( \alpha(x, y, z, i, j, k, f) \) as

\[
\exists i', j', k' \left( \phi_{e1}(x, y, z, i', j') \land \phi_{e2}(y, z, j', k') \land
\left( (i = i' \land j = j' \land k = 0_\Xi \land f = 0) \lor (i = 0_\Xi \land j = j' \land k = k' \land j = 1) \right) \right).
\] (3.1)

LET p, q, r, s, t, u, v, w, x, y, z;
f, i, j, k, t, φ, e1, e2, x, y, z.
' is %; Since “@” is a binary operator over Values”, 0_\Xi is 0.
(((u&v)&((x&y)&(z&(%q&%r))))&((u&w)&((y&z)&(%r&%s)))) & (((q=%q)&(r=%r))&((s=(t@t))&(p=(t@t))))+(((q=(t@t))&(r=%r))&((s=%s)&(p=(%t>#t))));

\textcolor{red}{FFFF FFFF FFFF FFFF (126)}
\textcolor{red}{FFFF FFFFF FFFF CFFFF ( 2)} \quad (3.2)

Appendix

Proof of Theorem 3. By induction on e.

Consider the expression \( e[K \cup K', V'] = \text{Ext}_f e_1[K, V] \), where \( f \) is of sort \((K, V) \rightarrow (K', V')\), and assume that \( \phi_{e_1}(\vec{x}_1, \vec{i}_1) \) is the formula obtained for \( e_1[K_1, V_1] \) by induction hypothesis. It is straightforward to see then that we can define

\[
\phi_e(\vec{x}_1, \vec{x}, \vec{i}) := \exists \vec{i}_1 \ ( \phi_{e_1}(\vec{x}_1, \vec{i}_1) \land R_f(\vec{x}_1, \vec{x}, \vec{i}, \vec{i}_1)).
\]

This finishes the proof of the theorem.

\[
\text{LET} \quad p, \quad q, \quad r, \quad s, \quad t, \quad u, \quad v, \quad w, \quad x:\n\quad \phi, \quad e, \quad R, \quad e_1, \quad i, \quad i_1, \quad x_1, \quad f, \quad x.
\]

\(((p&q)&(v&(x&u)))=((((p&s)&(v&%u))&((r&w)&((v&x)&(%u&t)))) \); 

\[
\begin{array}{cccccc}
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \quad (22) \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \quad (2) \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \quad (5) \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\end{array}
\]

\[
\text{Eqs. 3.2 and A3.2 as rendered are not tautologous. This refutes Th. 3 and by extension the Lara language as a unification of linear and relational algebra.}
\]