Refutation of second-order quantifier elimination and Craig interpolation in PIE

© Copyright 2020 by Colin James III All rights reserved.

Abstract: We evaluate two formulas for second-order quantifier elimination as not tautologous. This refutes the elimination of second-order quantifications in PIE and computing circumspection in Prolog. We evaluate the two examples of Craig interpolation (previously refuted elsewhere) as not tautologous to deny by extension PIE. We also evaluate the example of tableaux tool Graphviz as not tautologous to refute that. The broad results form a non tautologous fragment of the universal logic VL4.

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.)


Abstract. PIE is a Prolog-embedded environment for automated reasoning on the basis of first-order logic. Its main focus is on formulas, as constituents of complex formalizations that are structured through formula macros, and as outputs of reasoning tasks such as second-order quantifier elimination and Craig interpolation.

3 Second-order quantifier elimination in PIE

Second-order quantifier elimination is the task of computing for a given formula with second-order quantifiers, that is, quantifiers upon predicate or function symbols, an equivalent first-order formula. PIE so far just supports second order quantification upon predicate symbols, or predicate quantification.

Input: \( \exists p(\forall x(p(x) \rightarrow p(x)))\land \forall x(p(x) \rightarrow r(x))) \).
Result of elimination: \( \forall x(q(x) \rightarrow r(x)) \).

(3.1.1)

LET \( p, q, r, s, t, u, x: p, q, a, b, c, x. \)

\( (((q\&s)>(%p\&s))\&(\&(p\&s)>(r\&s)))=(((q\&#s)>(r\&#s)) ; \)

TTTT TTTT TCTT TCTT (3.1.2)

6 Computing circumscription as second-order quantifier elimination – PIE macros with Prolog bodies, result simplifications

The formula circ(p, p(a)) expands into: p(a)\( \land \neg \exists q(\forall x(q(x) \rightarrow p(x))) \land \forall x(p(x) \rightarrow q(x))) \). Second-
order quantifier elimination can be applied to compute the circumscription for the example:  
Input:  \( \text{circ}(p, p(a)) \);  Result of elimination:  \( p(a) \land \forall x(p(x) \rightarrow x=a) \).  
(6.1.1)

\[
((p \land s) \land \neg((q \land s) \land((q \land \neg x) \land(p \land \neg x))) \land ((p \land s) \land((p \land \neg x) \land(\neg x = s)))) = \\
((p \land s) \land((p \land \neg x) \land(\neg x = s)));
\]

\[
\begin{array}{cccccccc}
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\end{array}
\]

(6.1.2)

8 Craig interpolation  … Here is a propositional example:  Input:  \( p \land q \rightarrow p \lor r \).  Result of interpolation:  \( p \).  
(8.1.1)

\[
((p \land q) \land (p \lor r)) \rightarrow p ;
\]

\[
\begin{array}{cccccccc}
\text{F} & \text{T} & \text{T} & \text{F} & \text{T} & \text{T} & \text{T} & \text{T} \\
\end{array}
\]

(8.1.2)

Here is another example of Craig interpolation, where universal and existential quantification need to be combined:  Input:  \( \forall x \ (p(a,x) \land q \rightarrow \exists x \ (p(x,b) \lor r) \).  Result of interpolation:  \( \exists x \forall y \ (p(x,y) \).  
(8.2.1)

\[
(((p \land (s \land x)) \land q) \land((p \land (\neg x \land t)) \land (q \land r)) \rightarrow (p \land (\neg x \land \neg y))
\]

\[
\begin{array}{cccccccc}
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\end{array}
\]

(16)

\[
\begin{array}{cccccccc}
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\end{array}
\]

(16)

(8.2.2)

9 Further features of PIE  PIE supports the visualization of such tableaux as graph, rendered by the Graphviz tool. Here is an example:  Input:  \( \forall x \ (p(x) \land \forall x \ (p(x) \rightarrow q(x)) \rightarrow q(c) \).  Result of interpolation:  \( \forall x \ q(x) \).  
(9.1.1)

\[
(((p \land \neg x) \land((p \land \neg x) \land(q \land r))) \land ((q \land u) \land(q \land \neg x))
\]

\[
\begin{array}{cccccccc}
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\end{array}
\]

(16)

\[
\begin{array}{cccccccc}
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} & \text{F} \\
\end{array}
\]

(2)

(9.1.2)

Eqs. 3.1.2 and 6.1.2 as rendered are not tautologous. This refutes the elimination of second-order quantifications in PIE and computing circumspection in Prolog.

Eqs. 8.1.2 and 8.2.2 are not tautologous. This refutes Craig interpolation (previously refuted elsewhere) and by extension PIE here.

Eq. 9.1.2 is not tautologous. This further refutes tableaux tools such as Graphviz.