Refutation of subsumption demodulation in first-order theorem proving

© Copyright 2020 by Colin James III All rights reserved.

Abstract: We evaluate two equations in the example “to rewrite terms and simplify formulas using rewriting based on conditional equalities”, which are not equivalent and hence not tautologous. This refutes (subsumption) demodulation in first-order theorem proving, to include conjectured “reasoner” implementations of AVATAR, E, CVC4, SPASS, VAMPIRE, and Z3. These results 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, ∃, ∃, ◊, M; # necessity, for every or all, ∀, □, L;
(z=z) T as tautology, ⊤, ordinal 3; (z@z) ⊥ as contradiction, Ø, Null, ⊥, zero;
(%z>#z) N as non-contingency, Δ, ordinal 1; (%z<#z) ⊤ 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.


Abstract. Motivated by applications of first-order theorem proving to software analysis, we introduce a new inference rule, called subsumption demodulation, to improve support for reasoning with conditional equalities in superposition-based theorem proving. We show that subsumption demodulation is a simplification rule that does not require radical changes to the underlying superposition calculus. We implemented subsumption demodulation in the theorem prover VAMPIRE, by extending VAMPIRE with a new clause index and adapting its multi-literal matching component. Our experiments, using the TPTP and SMT-LIB repositories, show that subsumption demodulation in VAMPIRE can solve many new problems that could so far not be solved by state-of-the-art reasoners.

Remark 0.0: VAMPIRE is a “reasoner”, proof assistant in public domain Javascript of undocumented code without embedded test cases. It draws from AVATAR architecture of first-order provers as a “superposition prover” akin to “E, SPASS, CVC4, and Z3”, none of which we evaluate here but instead rely on VAMPIRE as emblematic, and symptomatic, of “[s]ubterm contextual rewriting” as “a refined notion of contextual rewriting”.

1 Introduction [ … In this paper we propose a generalized version of demodulation, called subsumption demodulation, allowing to rewrite terms and simplify formulas using rewriting based on conditional equalities … ]

Example 1. Consider the following formulas expressed in the first-order theory of integer linear arithmetic:

Remark 1.0: Symbol ≃ means equivalent here, and we ignore such repeated expressions.
\[ f(i) \equiv g(i), \ 0 \leq i < n \rightarrow P(f(i)) \quad (1.1.1.1) \]

\[
\text{LET} \quad p, q, r, s; \quad P, f, i, n.
\]

\[ \neg (#r < (s \cdot s)) < (s > (p \& (q \& \#r))) ; \quad \text{FFFF} \quad \text{FFFF} \quad \text{TTTT} \quad \text{CCCC} \quad (1.1.1.2) \]

Here, \( i \) is an implicitly universally quantified logical variable of integer sort, and \( n \) is integer-valued constant. First-order reasoners will first clausify formulas (1.1.1.1), deriving:

\[ f(i) \equiv g(i), \ 0 \leq i < n \lor P(f(i)) \quad (1.1.2.1) \]

\[ \neg (- (#r < (s \cdot s)) = (s = s)) + (- (#r < s) + (p \& (q \& \#r))) ; \quad \text{TTTT} \quad \text{TTTT} \quad \text{TTTT} \quad \text{TTTT} \quad (1.1.2.2) \]

As rendered, Eq. 1.1.1.2 is not equivalent to 1.1.2.2, hence refuting subsumption demodulation in first-order theorem proving of VAMPIRE and of other provers in that species.