Refutation of first-order reasoning by manipulating term algebras

Abstract: We evaluate two equations claimed as theorems but which are not tautologous. This refutes “a conservative extension of the theory of term algebras using a finite number of statements”, and hence first-order reasoning by manipulating term algebras. The fact that these equations are deemed tautologous by the Vampire prover, also refutes it as a prover. 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, $\mathcal{F}$ as contradiction, $\mathcal{N}$ as truthity (non-contingency), and $\mathcal{C}$ 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.)

Note for clarity, we usually distribute quantifiers onto each designated variable.


Abstract The theory of finite term algebras provides a natural framework to describe the semantics of functional languages. The ability to efficiently reason about term algebras is essential to automate program analysis and verification for functional or imperative programs over algebraic data types such as lists and trees. However, as the theory of finite term algebras is not finitely axiomatizable, reasoning about quantified properties over term algebras is challenging. In this paper we address full first-order reasoning about properties of programs manipulating term algebras, and describe two approaches for doing so by using first-order theorem proving. Our first method is a conservative extension of the theory of term algebras using a finite number of statements, while our second method relies on extending the superposition calculus of first-order theorem provers with additional inference rules. We implemented our work in the first-order theorem prover Vampire and evaluated it on a large number of algebraic data type benchmarks, as well as game theory constraints. Our experimental results show that our methods are able to find proofs for many hard problems previously unsolved by state-of-the-art methods. We also show that Vampire implementing our methods outperforms existing SMT solvers able to deal with algebraic data types.

4.2 Known results

Add two uninterpreted functions $+$ and $\cdot$ and consider the set $A$ of formulas defined as follows: [It is not hard to argue that in any extension of the $\Sigma$-algebra satisfying $A$, the functions $+$ and $\cdot$ are interpreted as the addition and multiplication on non-negative integers.]

$$\forall x \forall y \; (s(x) + y = s(x + y))$$

(4.2.2.1)

LET $p, q, r, s: p, x, y, s.$
\[(s\&#q)+#r)=(s\&(\#q+\#r))\); TTTT CCCC TTTT TTTT \hspace{1cm} (4.2.2.2)

\[\forall x \forall y (s(x) \cdot y = (x \cdot y) + y)\] \hspace{1cm} (4.2.4.1)

\[(s\&#q)&#r)=((\#q\&#r)+#r); TTTT CCCC TTTT TTCC \hspace{1cm} (4.2.4.2)\]

Eqs. 4.2.2.2 and 4.2.4.2 as rendered are not tautologous. This refutes “a conservative extension of the theory of term algebras using a finite number of statements”, and hence first-order reasoning by manipulating term algebras. The fact that these equations are deemed tautologous by the Vampire prover, also refutes it as a prover.