Refutation of the term rewriting approach to automated theorem proving

Abstract: Using three group axioms, two examples for an original and rewritten proofs are *not* tautologous. This refutes the term rewriting approach for automated theorem proving to form a *non* tautologous fragment of the universal logic VŁ4.

We assume the method and apparatus of Meth8/VŁ4 with `T`autology 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) **`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.

From: Hsiang, J.; Kirchner, H.; Lescanne, P.; Rusinowitch, M. (1992). The term rewriting approach to automated theorem proving. J. Logic Programming 1992:14:71-99. core.ac.uk/download/pdf/82371298.pdf

Example 2.1. … Consider the following set, called Group of axioms for groups:
x*e=x; x*i(x) =e; (x*y)*z=x*(y*z). ... An equational proof of e * x =x in groups is given below:

e*x=e*(x*e) =e*(x*(i(x)*i(i(x))))=e*((x*i(x))*i(i(x)))
=e*(e*i(i(x))) =(e*e)*i(i(x)) =e*i(i(x))
=(x*i(x))*i(i(x)) =x*(i(x)*i(i(x))) =x*e=x. (2.1.1)

LET p, q, r: e, I, x; & for *.

(((((p&r)=(p&(r&p)))=((p&(r&(q&r)))&(q&(q&r))))=(((p&((r&q)&r))&(q&(q&r)))=
(p&(p&(q&(q&r))))))=(((p&p)&(q&(q&r)))=(p&(q&(q&r))))) =
(((((r&q)&r)&(q&(q&r)))=((r&(q&r))&(q&(q&r))))=((r&p)=r)) ;
FFFF `T`**F**`TT` **FFFF** `T`**F**`TT` (2.1.2)

Example 2.2. For instance, x*e→x; x*i(x)→e; (x*y)*z→x*(y*z) is a rewrite system. Rewriting a term with a rewrite system R consists in replacing a subterm which matches a left-hand side of a rewrite rule by the right-hand side whose variables are bound to values computed by the matching algorithm. This relation is denoted by $\rightarrow_R$. Iterating this process is called reducing. If two terms can be reduced to a same one, a special equational proof is obtained, called a rewrite proof. A term which cannot be rewritten is said to be in normal form.

Example 2.3. The rewrite rules of Example 2.2 are used in the equational proof of Example 2.1 as follows:

e*x←e*(x*e)←e*(x*(i(x)*i(i(x))))←e*((x*i(x))*i(i(x)))
→e*(e*i(i(x)))←(e*e)*i(i(x))→e*i(i(x))
←(x*i(x))*i(i(x))→x*(i(x)*i(i(x)))→x*e→x. (2.3.1)

(((((p&r)<(p&(r&p)))<((p&(r&(q&r)))&(q&(q&r))))<(((p&((r&q)&r))&(q&(q&r)))>
(p&(p&(q&(q&r))))))<(((p&p)&(q&(q&r)))>(p&(q&(q&r))))) <
(((((r&q)&r)&(q&(q&r)))>((r&(q&r))&(q&(q&r))))>((r&p)>r)) ;
FFFF FFFF FFFF FFFF (2.3.2)

Obviously it is not a rewrite proof. There are peaks, i.e., terms from which issue two sequences of rewritings, and valleys, i.e., terms where rewriting is not applied any more. Such terms, for instance e * x, e *(e * i(i(x))), e * i(i(x) and x are in normal form for the rewrite system.

Eqs. 2.1.2 and 2.3.2 as rendered are *not* tautologous. This refutes the term rewriting approach for automated theorem proving.