Refutation of Isabelle/HOL

Abstract: Eight of eleven equations are evaluated as *not* tautologous. This means the rewrite-engine or simplifier tool is not confirmed, the conjunction is not effectively defined by three rules, and other reasoning steps are not expressed similarly, hence refuting Isabelle/HOL. These anomalies 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) **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: Paulson, L.C.; Nipkow, T.; Wenzel, M. (2019). From LCF to Isabelle/HOL. arxiv.org/pdf/1907.02836.pdf

3. Isabelle in the early days: a logical framework
Isabelle originated in a project to build an LCF-style proof [logic for computable functions] assistant for Martin-Löf's constructive type theory.

A special case of unification is matching where the variables of only one of the two terms are instantiated. Isabelle's rewrite engine (aka the simplifier) is based on higher-order pattern matching. Thus the simplifier can deal with many standard transformations of quantified terms, for example the following:

$$(\forall x.\ P(x) \wedge Q(x)) = (\forall x.\ P(x)) \wedge (\forall x.\ Q(x)) \qquad (3.1.1)$$

LET p, q, r, s: P, Q, x, t.

((p&#r)&(q&r))=((p&#r)&(q&#r)) ;
TTTT TTTT TTTT TTTT (3.1.2)

$$(\forall x.\ P \vee Q(x)) = P \vee (\forall x.\ Q(x)) \qquad (3.2.1)$$

((#r&p)+(q&r))=(p+(q&#r)) ; TFTF TNNT TFTF TNNT (3.2.2)

$$(\forall x.\ x = t \wedge P(x)) = P(t)) \qquad (3.3.1)$$

((#r=s)&(p&r))=(p&s) ; TTTT TNTN TFTF TNTN (3.3.2)

It appears that Isabelle was the first theorem prover to support higher-order rewrite rule.

5. Automation

5.1. The classical reasoner

As mentioned in section 3 above, Isabelle supported both unification and backtracking from the start, with the aim of incorporating ideas from first-order automatic proof procedures. In the context of interactive proof, unification provided the ability to prove a subgoal of the form $\exists x.\varphi(x)$ by removing the quantifier and proving $\varphi(?t)$, where ?t stood as a placeholder for a concrete term to be supplied later. Through unification, this term could even be built up incrementally. Dually, unification provided a means of using a universally quantified fact $\forall x.\varphi(x)$, when the required instances were not immediately obvious.

Simple automation is achievable through a combination of obvious applications of the propositional connectives ($\wedge$, $\vee$, $\neg$, etc.) along with heuristics for performing quantifier reasoning. Stronger automation is obtainable by borrowing well-known techniques for classical first-order logic theorem proving. But the most important idea is to embrace the concepts of natural deduction in application theories as well as in pure logic. Natural deduction prefers the use of simple inference rules focusing on a single symbol.

For example, conjunction is effectively defined by the following three rules:

$(\varphi > \psi) > (\varphi \wedge \psi)$ (5.1.1)

LET p, q: φ, ψ.

(p>q)>(p&q) ; **F**T**F**T **F**T**F**T **F**T**F**T **F**T**F**T (5.1.2)

$(\varphi \wedge \psi) > \varphi$ (5.2.1)

(p&q)>p ; TTTT TTTT TTTT TTTT (5.2.2)

$(\varphi \wedge \psi) > \psi$ (5.3.1)

(p&q)>q ; TTTT TTTT TTTT TTTT (5.3.1)

The intersection of two sets has a technical definition that would greatly complicate reasoning, but it is easy to derive inference rules for intersection in the style of natural deduction (and analogous to those above):

$((a \in A) > (a \in B)) > a \in A \cap B$ (5.4.1)

LET p, q, r, s: a, b, A, B

((p<r)>(p<s))>(p<(r&s)) ; **F**T**F**T **F**T**F**T **F**T**F**T **FFFF** (5.4.2)

$(a \in A \cap B) > a \in A$ (5.5.1)

(p<(r&s))>(p<r) ; TTTT T**F**T**F** TTTT TTTT (5.5.2)

$(a \in B \cap A) > a \in B$ (5.6.1)

(p<(s&r))>(p<s) ; TTTT TTTT T**F**T**F** TTTT (5.6.2)

Many other reasoning steps can be expressed similarly:

$$((A \subseteq B) > (a \in A)) > a \in B \tag{5.7.1}$$

LET p, q, r, s: a, A, B, C.

$$(\sim(r<q)>(p<q))>(p<r)\ ; \qquad \texttt{TTTT } \mathbf{FT}\texttt{TT TTTT } \mathbf{FF}\texttt{TT} \tag{5.7.2}$$

$$((A \subseteq B) > (B \subseteq C)\ ) > A = B \tag{5.8.1}$$

$$(\sim(r<q)>\sim(s<r))>(r=s)\ ; \qquad \texttt{TTTT } \mathbf{FFFF}\texttt{ TTTT TTTT} \tag{5.8.2}$$

Eqs. 3.2.2, 3.3.2, 5.1.2, 5.4.2, 5.5.2, 5.6.2, 5.7.2, and 5.8.2 as rendered are *not* tautologous. This means the rewrite engine or simplifier tool is not confirmed, the conjunction is not effectively defined by three rules, and other reasoning steps are not expressed similarly, hence refuting Isabelle/HOL.