Refutation of tactic learning and proving for the Coq proof assistant

Abstract: We evaluate a claimed theorem for proof assistant Coq which is not tautologous. This refutes the instant tactic language for Coq. What follows is that the claim of tactic learning by Coq and Gallina is also denied. 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) 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.


Abstract We present a system that utilizes machine learning for tactic proof search in the Coq Proof Assistant ...

1 Introduction The Coq Proof Assistant .. is an Interactive Theorem Prover in which one proves lemmas using a tactic language. ... (1) Decision procedures only apply to particular domains, requiring the user to know when they are appropriate, and (2) Coq’s search tactics require careful construction of hint databases to be performant. The system we present in this paper learns from previously written proof scripts and how they are applied to proof states. This knowledge can then be used to suggest a tactic to apply on a previously unseen proof state or to perform a full proof search and prove the current lemma automatically. Learning on the tactic level has some advantages over learning from low-level proof terms in Gallina, Coq’s version of the Calculus of Inductive Constructions .. : (1) Tactics represent coarser proof steps than the individual identifiers in a proof term. They are also more forgiving; a minor modification to a tactic script is more likely to be nondestructive than a minor modification to a proof term, making the machine learning task easier. (2) By learning on the tactic level, we allow the user to introduce domain-specific knowledge to the system by writing custom tactics. By using these tactics in hand-written proofs, the system will automatically learn of their existence and start to use them.

From: Delahaye, D. (2014). A tactic Language for the system Coq. researchgate.net/publication/220896176_A_Tactic_Language_for_the_System_Coq/link/09e415127430e7f06c000000/download

Abstract. We propose a new tactic language for the system Coq, which is intended to enrich the current tactic combinators (tacticals). This language is based on a functional core with recursors and matching operators for Coq terms but also for proof contexts. It can be used directly in proof scripts or in toplevel definitions (tactic definitions). We show that the implementation of this language involves considerable changes in the interpretation of proof scripts, essentially due to the matching
operators. We give some examples which solve small proof parts locally and some others which deal with non-trivial problems. Finally, we discuss the status of this meta-language with respect to the Coq language and the implementation language of Coq.

1 Introduction In a proof system, we can generally distinguish between two kinds of languages: a proof language, which corresponds to basic or more elaborate primitives and a tactic language, which allows the user to write [one's] own proof schemes.

... But, now suppose that we want to show that the set of natural numbers has more than two elements. This can be expressed as follows:

\[ |- \ (\exists x : N. \exists y : N. \forall z : N. x = z \lor y = z) \rightarrow \bot \quad (1.1.1) \]

\[
\text{LET } p, q, r, s : N, x, y, z. \\

(((\%q&(p&%r))&(p&\#s)&(p&\%q)))=((\#s+\%r)=\#s))>(s@s) ; \\
NNNN \text{ FFFF} NNNN \text{ FFFN} \\
\]

Remark 1.1.2: Eq. 1.1.2 as rendered is not tautologous. This refutes the instant tactic language for Coq because as non-theorem is asserted as a theorem. What follows is that the claim of tactic learning by Coq and Gallina is also denied.