Coq au vin
The Coq proof assistant and the Curry-Howard correspondence

Juan Pedro Villa-Isaza
jvillais@eafit.edu.co

June 8, 2011

Abstract

The *Coq* ([3]) proof assistant is a formal proof management system that is based on the Curry-Howard (*propositions-as-types*, *formulas-as-types*, *proofs-as-programs*) correspondence. *Coq au vin* (literally, "cock in wine") analyzes some aspects of this correspondence (propositional logic, predicate logic, and proof irrelevance) and provides some examples of "proofs viewed as programs." In addition, it compares Coq with the *Agda* proof assistant through the Curry-Howard correspondence.

1 Contents

1.1 Introduction

This section introduces the basic concepts of the Coq proof assistant and a very brief summary of the Curry-Howard correspondence.

1.2 Propositional logic

The first part of this section is a simple introduction to the Coq proof assistant. It presents some of the basic ideas used for proving theorems with Coq and applies them for proving the transitivity of the implication (for a complete analysis of this example, see [2]).

The second part studies the implementation of the introduction and elimination rules of four propositional connectives (implication, conjunction, disjunction, and absurdity), and the definitions of three additional connectives (`True`, negation, and bi-implication).

The source code contains proofs for the examples in [10] and some exercises in [2] and [11].

1

1.3   Predicate logic

This section presents the basic Coq definitions of the universal quantification and the existential quantification. The source code contains proofs for three examples in [10].

1.4   Proof irrelevance

The last section of the presentation is a brief analysis of an important difference between building proofs and programs (*proof irrelevance*). It contains a simple example based on Coq's standard library.

2   Coq au vin

For the slides of *Coq au vin*, see `slides/coqav-slides.pdf`. For the Coq source code, which contains all the examples and exercises, see `src/coqav.v`. And for the Agda source code, see `src/Coqav.agda`.

This file (`doc/coqav.pdf`) contains a brief summary of *Coq au vin* and the complete list of references that were used one way or another. [2], [11], and [10] are the main sources for the analysis.

References

[1] Yves Bertot. *Coq in a Hurry*. INRIA, April 2010. Version 5. `http://cel.archives-ouvertes.fr/inria-00001173`.

[2] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[3] The Coq Development Team. The Coq Proof Assistant. Action for Technological Development (ADT) Coq. `http://coq.inria.fr/`.

[4] The Coq Development Team. *The Coq Proof Assistant Reference Manual.* INRIA, April 2011. Version 8.3pl2. `http://coq.inria.fr/refman/`.

[5] Eduardo Giménez and Pierre Castéran. *A Tutorial on [Co-]Inductive Types in Coq*, August 2007. `http://www.labri.fr/Perso/~casteran/RecTutorial.pdf`.

[6] Gérard Huet, Gilles Kahn, and Christine Paulin-Mohring. *The Coq Proof Assistant. A Tutorial*, April 2011. Version 8.3pl2. `http://coq.inria.fr/V8.1/files/doc/Tutorial.pdf`.

[7] Joan Moschovakis. Intuitionistic logic. In Edward N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, summer 2010 edition, 2010. `http://plato.stanford.edu/archives/sum2010/entries/logic-intuitionistic/`.

[8] Stewart Shapiro. Classical logic. In Edward N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, winter 2009 edition, 2009. `http://plato.stanford.edu/archives/win2009/entries/logic-classical/`.

[9] Andrés Sicard-Ramírez. Proofs = Programs. Días de la ciencia aplicada (EAFIT), September 2009. `http://www1.eafit.edu.co/asicard/slides/chi-slides.pdf`.

[10] Andrés Sicard-Ramírez. Dependently typed functional languages (CB-0683). EAFIT, 2011. `http://www1.eafit.edu.co/asicard/teaching/dtfl-CB0683/`.

[11] Morten Heine Sørensen and Paweł Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2006.

[12] Dirk van Dalen. *Logic and Structure*. Universitext. Springer, fourth edition, 2004.

[13] Jan von Plato. The development of proof theory. In Edward N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, spring 2009 edition, 2009. `http://plato.stanford.edu/archives/spr2009/entries/proof-theory-development/`.