Practising logic through the web

Freek Wiedijk

Institute for Computing and Information Sciences
Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

Abstract. We present the ProofWeb system for practising natural deduction in predicate logic and for remotely working with the Coq proof assistant. The ProofWeb system can be used for free, both for trying it out as a guest as well as for hosting computer labs for logic and proof assistant courses, on the Nijmegen ProofWeb server `http://proofweb.cs.ru.nl/`.

A computer lab for a logic course

Suppose you are a lecturer at a computer science department who has to teach the introductory logic course. That course teaches natural deduction in three standard logics:

- Propositional logic
- Predicate logic
- Predicate logic with equality

You would like the computer to be used by your students for their logic exercises. What should you use?

In the Netherlands many of these courses currently use the Jape system from the UK [3]. It allows students to check simple deductions with a computer program that has a pleasant point-and-click interface. However, the Jape system has restrictions. It is not possible to save an unfinished deduction and later continue working on it. It is not possible to have types for variables. It is not possible to reason about time in the deductions, or to reason arithmetically in other contexts for that matter. This all makes the use of Jape inconvenient for larger examples. Also, with Jape there is no easy way to run a course in an centrally organized manner. Each student has to install Jape on their own system, it is hard to keep track of what the students are doing with it, and grading students' work is labor intensive.

Here we present the ProofWeb system, as a better system to be used for computer labs for logic courses. ProofWeb was developed as a joint effort between the Radboud University Nijmegen and the Free University Amsterdam, in a small education innovation project funded by the Surf Foundation called *Web-deductie voor het onderwijs in formeel denken*. ProofWeb already has been, and still is, used in about a dozen logic courses in various universities.

The Coq proof assistant as a web application

Proof assistants are systems for developing and checking deductions on the computer. These systems are both used for verification of the correctness of software and hardware, as well as for checking the correctness of mathematics. There are several proof assistants to choose from (important examples are PVS, HOL, Isabelle, ACL2, The B Method, Twelf, and in some sense even Jape). One of the best proof assistants currently available is the Coq system developed at INRIA in France [4, 1]. Coq has been used for many impressive projects, like the validated Compcert C compiler which compiles a large subset of C to assembly for the PowerPC family of processors, getting a performance similar to the gcc compiler with the first level of optimization turned on [10, 2], and the formal proof of the Four Color Theorem by Georges Gonthier [6].

Before the ProofWeb project, Henk Barendregt had for a long time already been asking for a web interface to the Coq system, to allow people to work with Coq without first having to go through the trouble of installing it on their computer. There had been a project in which mathematicians from Russia were supposed to contribute (they would be paid a fixed price for each lemma proved), but before they had Coq running correctly on their systems the project already was over. If a Coq environment had been available on the web this might have gone differently. But Henk wanted a web interface to Coq for a more ambitious goal: to have the whole world help with encoding all of mathematics in Coq. This would be an important step to make that come about.

Cezary Kaliszyk, one of the PhD students in Henk's group in Nijmegen, happened to be an expert on Web interfaces. His PhD research was on making proof assistants more friendly, but his master's thesis had been about Web interfaces. When he heard about Henk's wish, in a Christmas holiday he whipped together a simple but nice web interface to the Coq system. This interface later was extended into the ProofWeb system. Although various people were involved in that project, Cezary remained the sole developer of the system.

The ProofWeb system

The ProofWeb system has the following distinctive properties:

- With ProofWeb the students work with the Coq proof assistant. Their input is not being pre-processed by the ProofWeb interface; i.e., the text that ProofWeb users are typing are actual Coq proof scripts. With ProofWeb the students are working with an industrial strength interactive theorem prover, with the full power of that system available to them from the start.
- ProofWeb shows deductions in a style that matches as closely as possible the deductions the way they are presented in elementary logic courses. Although ProofWeb shows the Coq presentation of the state of the proof, there is also a display that really looks like the diagrams from logic textbooks. In fact, the ProofWeb system intentionally was made fully compatible with a good logic textbook: *Logic in Computer Science: Modelling and Reasoning about*

Systems by Michael Huth and Mark Ryan [7]. If a course uses this book then ProofWeb is a good choice for the lab work of that course. And if a course wants to use ProofWeb for the lab work, then this book is a good textbook to be used for the non-computer part of the course.
- ProofWeb comes with an extensive manual [8] that both summarizes natural deduction for predicate logic and presents all the details of working with the ProofWeb system. This manual can be downloaded as a PDF file from the ProofWeb web site.
- ProofWeb comes with a collection of more than a hundred simple logic exercises to be worked with the ProofWeb system. Courses can have their own set of exercises, but this set is a good default choice for an introductory logic course.
- To use ProofWeb there is no need to install special software. In fact ProofWeb does not even use a plug-in. All that one needs is a compatible web browser. ProofWeb users can access ProofWeb from anywhere on the internet. (For example, if necessary, students might go into an internet café to finish their homework.)
- ProofWeb has good interfaces for both students and teachers to manage their courses. Students will see a list of exercises, and the status of these exercises. Teachers will see a list of students, and the status of those students.

ProofWeb for teaching Coq

ProofWeb can be used for teaching logic to undergraduate students, but it also can be used for teaching proof assistants to graduate students. In fact, about half of the current ProofWeb courses are type theory courses that teach the Coq system.

In such courses the exercises do not have the shape of having to prove a single first order formula. Instead they are long files with many lemmas, where the students have to fill in the proofs of those lemmas. Generally students then will have to complete one such an exercise per week.

Fitch-style natural deduction

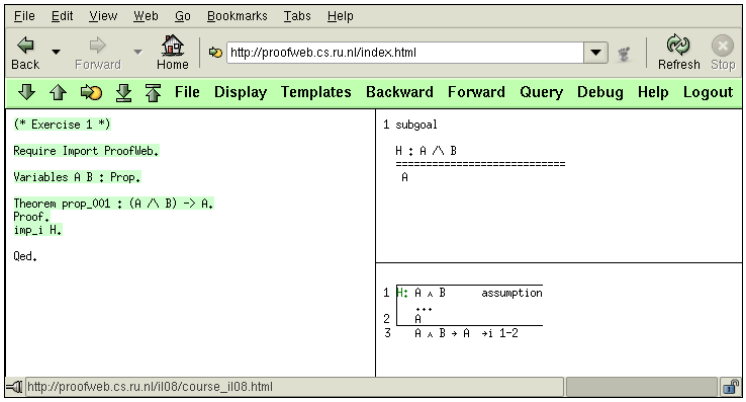
There are two styles of presenting natural deduction proofs on paper. The more commonly taught style was introduced by Frederic Fitch. These proofs consists of lines that are grouped together either by boxes around them or by having 'flags' with assumptions in the margin. (Using flags is typical of the way this kind of proof is taught at the Technical University of Eindhoven. This style originated in the Automath project from the sixties and seventies.)

Fitch-style proofs are the style of natural deduction presented in the book by Huth and Ryan that ProofWeb especially was designed to match. However, Fitch-style proofs do not exactly have the structure of Coq proofs. For this reason, in the ProofWeb project one of the major difficulties was to get Fitch-style deduction to work well with the Coq system [9].

Here is a Fitch-style proof as shown in ProofWeb:

1	H1:	∃x, (P x ∨ ¬Q a)	assumption
2	H2:	Q a	assumption
		b	
3	H3:	P b ∨ ¬Q a	assumption
4	H4:	P b	assumption
5		∃x, P x	∃i 4
6	H5:	¬Q a	assumption
7		⊥	¬e 6,2
8		∃x, P x	⊥e 7
9		∃x, P x	∨e 3,4-5,6-8
10		∃x, P x	∃e 1,3-9
11		Q a → ∃x, P x	→i 2-10
12		∃x, (P x ∨ ¬Q a) → Q a → ∃x, P x	→i 1-11

When doing the proof, this picture will be growing in the lower-right pane of the ProofWeb window. Here is what the ProofWeb interface looks like halfway down a simpler example:



Both the Coq view of the proof as well as the textbook display of the proof are present simultaneously. To our surprise when working on exercises like this, students mostly completely ignored the Coq proof state in the upper-right pane and just looked at the lower-right.

Gentzen-style natural deduction

The other style of natural deduction that is commonly used was introduced by Gerhard Gentzen. This is the style that is used in the textbook *Logic and Structure* by Dirk van Dalen [5]. (ProofWeb is not completely compatible with this book – we decided that internal consistency was important, so we had our Gentzen-style proof display be influenced by the Huth and Ryan conventions – but it comes very close.)

In the case of Gentzen-style natural deduction the proofs look like trees, growing upward from the statement that is proved. When doing this kind of

proof on paper one generally gets space problems fast, but in a computer one does not have this problem, as in that case one has scroll bars.

Here is the same example proof, but this time presented in Gentzen style:

$$\dfrac{\dfrac{[\exists x,\ (P\ x \lor \neg Q\ a)]^{H1} \qquad \dfrac{[P\ b \lor \neg Q\ a]^{H3} \qquad \dfrac{[P\ b]^{H4}}{\exists x,\ P\ x}\exists i \qquad \dfrac{\dfrac{[\neg Q\ a]^{H5} \quad [Q\ a]^{H2}}{\bot}\neg e}{\exists x,\ P\ x}\bot e}{\exists x,\ P\ x}\lor e[H4,H5]}{\exists x,\ P\ x}\exists e[H3]}{Q\ a \to \exists x,\ P\ x}\to i[H2]}{\exists x,\ (P\ x \lor \neg Q\ a) \to Q\ a \to \exists x,\ P\ x}\to i[H1]$$

The ProofWeb code that generated both proofs (which just are different displays of the same proof) was:

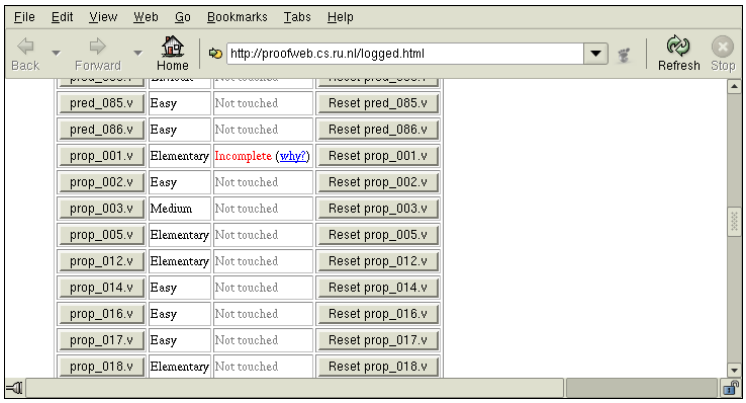
```
Require Import ProofWeb.

Variable P Q : D -> Prop.
Variable a : D.

Theorem example :
  exi x, (P(x) \/ ~Q(a)) -> Q(a) -> exi x, P(x).
Proof.
imp_i H1.
imp_i H2.
exi_e (exi x, (P(x) \/ ~Q(a))) b H3.
exact H1.
dis_e (P(b) \/ ~Q(a)) H4 H5.
exact H3.
exi_i b.
exact H4.
fls_e.
neg_e (Q(a)).
exact H5.
exact H2.
Qed.
```

This is an actual Coq input script. The commands occurring in this script can be selected from menu's in the ProofWeb interface, so the students do not need to know these commands by heart. Also, these commands are extensively explained in the ProofWeb manual.

The student's view of ProofWeb

When a student follows a course that uses the ProofWeb system, he or she will go to the ProofWeb server that hosts the course. The student will then select the

course from a menu, login in to the system by entering username and password, and then will be presented with the list of exercises:



Each exercise will have four possible statuses, which are color coded:

Gray	Not touched
Red	Incomplete
Orange	Correct
Green	Solved

The goal for the student will be to work the exercises until the traffic lights in his list are all green. (The orange status means that the proof is correct, but that the student used proof steps that are not allowed for the course. For example Coq's powerful automation will not be allowed when doing exercises in elementary logic.)

The teacher's view of ProofWeb

The teacher can login for the course too, but with the teacher's interface. He or she then will be presented with a table that lists all the students, with for each student a count of how many of that student's exercises there are of each of the four different colors. The teacher also can login as if he were a specific student, to grade exercises or maybe help the student with finding a solution.

Apart from this, the teacher interface also allows the teacher to add student logins or change passwords. Finally there is a button for downloading all files for the course as a big tar file, for archival purposes at the end of a course.

The MathWiki project

ProofWeb was primarily developed for logic education, but in Nijmegen we have more ambitious goals for it.

Recently, the project *MathWiki: a Web-based Collaborative Authoring Environment for Formal Proofs* was funded in NWO's vrije competitie, to develop a

system to be called MathWiki. This system will be a cross between a Wikipedia for mathematics, and the system that Henk was dreaming of where all the world would help build a Coq version of all of mathematics. An aspect of the MathWiki project is that it is not supposed to be just about the Coq proof assistant. Eventually many proof assistants will be available through our interface.

A prototype of MathWiki already was developed and a web page about this work can be found on the ProofWeb server.

Trying it?

If you want to try ProofWeb: it is completely free. Currently there are three way of using the system:

- First you can access it as a guest user. For this you do not even need to register. Just click the Guest login button. It is probably useful to first look through the ProofWeb manual to know what to do next.
- Second you can host courses on the Nijmegen ProofWeb server. For this, just send an email message to `proofweb@cs.ru.nl`.
- Third, you might not trust someone else with your students' data. In that case you might download the ProofWeb server and install it on a machine of your own. At the moment this has not been done much, and you will probably need some help from Nijmegen with that, which of course we will be happy to provide.

We hope ProofWeb will be useful both for logic teaching, as well as for exposing more students to proof assistants. If you have any questions about ProofWeb, just send mail to

`proofweb@cs.ru.nl`

Or you should surf to

`http://proofweb.cs.ru.nl/`

and take a look for yourself at the system that we developed.

References

1. Yves Bertot and Pierre Castéran. *Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.
2. Sandrine Blazy, Zaynah Dargaye, and Xavier Leroy. Formal Verification of a C Compiler Front-end. In *FM 2006: Int. Symp. on Formal Methods*, volume 4085 of *Lecture Notes in Computer Science*, pages 460–475. Springer, 2006.
3. Richard Bornat and Bernard Sufrin. Jape's Quiet Interface. In N. Merriam, editor, *User Interfaces for Theorem Provers (UITP '96)*, Technical Report, pages 25–34. University of York, 1996.
4. Coq Development Team. *The Coq Proof Assistant Reference Manual*, 2008.
5. Dirk van Dalen. *Logic and Structure*. Springer, 4th edition, 2004.

6. Georges Gonthier. A computer-checked proof of the Four Colour Theorem. ⟨`http://research.microsoft.com/~gonthier/4colproof.pdf`⟩, 2006.
7. Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2nd edition, 2004.
8. Cezary Kaliszyk, Femke van Raamsdonk, Freek Wiedijk, Hanno Wupper, Maxim Hendriks, and Roel de Vrijer. Deduction using the ProofWeb system. Technical Report ICIS-R08016, Institute for Computing and Information Sciences, University of Nijmegen, 2008.
9. Cezary Kaliszyk and Freek Wiedijk. Merging procedural and declarative proof. To be published in the proceedings of TYPES 2008, 2009.
10. Xavier Leroy. Formal Certification of a Compiler Back-end, or: Programming a Compiler with a Proof Assistant. In *POPL'06*, Charleston, South Carolina, USA, 2006.