Teaching logic using a web interface for Coq

October 31

Cezary Kaliszyk
Radboud University Nijmegen
cek@cs.ru.nl
Presentation Plan

- Web Interface
  - Motivation
  - Requirements
  - Architecture
  - Efficiency and Security

- Teaching Logic
  - Tactics
  - Graphical presentation of proofs
  - Problem set
Why the Web?

- Popular - No installation or configuration
- e-mail clients, calendars, maps, chats, word processing, ...
- wikis and Wikipedia
- some tools for proofs
Web Technologies

- Plugins: Java, Flash
- JavaScript
- DOM
- XmlHttp
- Asynchronous DOM modifications
  - sometimes called AJAX or Web Application
Proof Assistants

- Often complicated to install
- Proofs are developed locally
  - Versioning systems
- Static web pages are generated to display proofs on the web
  - tactic-mode proofs
Architecture (1/2)

- Lightweight client part in browser
  - User does not need to install anything
- Specialized web-server
  - Prover sub-processes
- Minimal communication
Teaching logic using a web interface for Coq

Architecture (2/2)

User of ProofWeb

Web Browser

Presented page

handling of keypresses and clicks

JavaScript

XmlNode

DOM

Callback

User’s Session

User’s Prover

User’s Session

User’s Prover

Web Server
Implementation of a prototype

- Client part: 30kB of JavaScript and HTML
- Server part: 800 lines of OCaml code, uses OCamlHttpd runs prover subprocesses
- Tested with Mozilla based browsers, Internet Explorer and Opera
- On any platform/architecture one can easily access the interface
  - No java, plugins installations or privileges are required
User Security and Efficiency

- System and browser errors
- Efficiency of an interpreted language
  - Browser efficiency
- Network latency
  - TCP Ping time
Server Security

- Crackers, “Too-powerful” provers
- Availability of work and files and equal sharing of all resources
- The communication mechanism
- Compilation and dependencies
Server Security

- Crackers, “Too-powerful” provers
  - Sandboxing

- Availability of work and files and equal sharing of all resources
  - disk quota, CPU quota, memory quota

- The communication mechanism
  - HTTPS

- Compilation and dependencies
Server Efficiency

- Depends on:
  - Number of users, server configuration, provers, usage of automated techniques, ...
- Possible to start provers on different machines
- We expect compilation of dependencies to be the main bottleneck in bigger projects (more in Pierre’s talk)
Project parts

- programming the interface
- tactics for first-order logic
- graphical presentation of proofs
- a problem set
- course notes / manual
Example of a tactic

disjunction elimination:

Ltac dis_el X H1 H2 :=
    match X with
    | ( _ / _ ) =>
        assert X;
        [ idtac |
            match goal with
                | x : X |- _ =>
                    elim x; [intro H1 | intro H2]; clear x
            end
        ]
    | _ => fail "The first argument is not a disjunction"
end.
Example of an exercise

Theorem exercise_024 : (A \or B) \or \neg A \rightarrow B.
Proof.

imp_in z.
dis_el (A \or B) y1 y2.
con_ell (\neg A).
ass z.
neg_el A.
con_elr (A \or B).
ass z.
ass y1.
ass y2.
Qed.
Example of a proof tree

\[(A \lor B) \land \neg A \quad \neg A \quad \neg \text{ass}[y1] \quad \text{ass}[z] \]

\[(A \lor B) \land \neg A \quad \text{E}(l) \land \text{E}(r) \land \neg \text{ass}[y1] \quad \text{ass}[z] \]

\[(A \lor B) \land \neg A \quad \text{E}(r) \land \neg \text{ass}[y1] \quad \text{ass}[z] \]

\[\text{A} \lor \text{B} \]

\[\text{Ev}[y1, y2] \]

\[\text{B} \]

\[\text{B} \]

\[(A \lor B) \land \neg A \rightarrow \text{B} \]

\[I \rightarrow [z] \]
Example of a Fitch proof tree

\[
\begin{align*}
H: & \quad (\forall x:D, P x \rightarrow Q x) \land (\exists x:D, P x) \quad \text{assumption} \\
\exists & x \quad P x \\
\forall y:D, P y \rightarrow Q y & \quad \text{assumption} \\
P x & \rightarrow Q x \\
Q x & \\
\exists x:D, Q x & \\
P x \rightarrow \exists x0:D, Q x0 & \\
\forall x:D, P x \rightarrow \exists x0:D, Q x0 & \quad \forall i 2-7 \\
& \quad \vdots \\
(\forall x:D, P x \rightarrow Q x) \land (\exists x:D, P x) & \\
\exists x:D, P x & \\
\exists x:D, Q x & \quad \exists e 8,10 \\
(\forall x:D, P x \rightarrow Q x) \land (\exists x:D, P x) \rightarrow \exists x:D, Q x & \quad \exists i 1-11
\end{align*}
\]
Proofweb in practice

The system has been and will be used in two types of courses so far:

- **Graduate courses:**
  - Logical Verification (Amsterdam)
  - Type Theory (Nijmegen)
  - Master Class on Type Theory and Proof Assistants (Nijmegen)

- **Undergraduate courses:**
  - Beweren en Bewijzen (Nijmegen)
  - Introduction to logic (Amsterdam)
Coq (with Proofweb) vs. Jape (or cousins)

- Prepares for later use of Coq
- Message to students: no plaything
- A broader system (CNF, satisfiability, logic with numbers, modal logic)
- No random proving by clicking
- Saving (incomplete) proofs is possible
- Centralized architecture
Work in progress

The needed infrastructure for creating a Wiki.

- Many provers and versions of provers
- Security Policy
- Teacher interface vs side utils in Wikis
- Standard protocol