

OBLICZENIA I WNIO SKOWANIE W SYSTEMIE CoQ

Małgorzata Biernacka

Instytut Informatyki UWr

Wykład 1
26.02.2013

TREŚĆ I FORMA ZAJĘĆ

- ▶ praktyczne wprowadzenie do Coq'a, z niezbędnym minimum teorii
- ▶ zaliczenie i ocena z pracowni na podstawie list zadań (ok. 5-6)
- ▶ nie ma egzaminu
- ▶ zaliczenie i ocena z wykładu na podstawie wykonanego projektu lub prezentacji na zajęciach
- ▶ listy zadań można oddawać **tylko** w czasie pracowni (poza przypadkami uzasadnionymi chorobą itp.)
- ▶ bieżące ogłoszenia, materiały i listy zadań:
<http://www.ii.uni.wroc.pl/~mabi/coq2013>

MATERIAŁY

- ▶ coq.inria.fr – punkt wyjścia do eksploracji zasobów Coqa
- ▶ [Reference Manual](#)
- ▶ A. Chlipala, *Certified programming with dependent types* – bardzo dobry podręcznik
- ▶ podręcznik do przedmiotu *Software Foundations* w Uni. of Pennsylvania (www.cis.upenn.edu/~bcpierce/sf)
- ▶ Y. Bertot, P. Casteran *Coq'Art* – książka przydatna jako podręcznik, ale przestarzała

ŚRODOWISKO PRACY

- ▶ instalujemy wersję 8.4pl1 ze strony <http://coq.inria.fr/download>
- ▶ do interaktywnej pracy z Coqiem używamy środowiska CoqIDE lub trybu ProofGeneral dla emacsa
- ▶ czasem będziemy potrzebować starszych wersji Coqa (dostępne na coq.inria.fr)

CZYM JEST CoQ

proof assistant/proof management system: narzędzie do interaktywnego dowodzenia twierdzeń

- ▶ system ogólnego przeznaczenia
- ▶ *język specyfikacji* umożliwia bezpośrednie modelowanie problemu
- ▶ w języku specyfikacji możemy zapisywać matematyczne definicje, twierdzenia, dowody, a także wykonywalne programy funkcyjne
- ▶ interaktywne środowisko dowodzenia twierdzeń pozwala na dowodzenie za pomocą *języka taktyk*
- ▶ słaba *automatyzacja* dowodzenia
- ▶ umożliwia programowalną automatyzację dowodzenia
- ▶ możliwa ekstrakcja programów z dowodów
- ▶ inne systemy o podobnym przeznaczeniu: HOL Light, Isabelle/HOL, Mizar, PVS, Twelf, Matita

DO CZEGO SŁUŻY COQ

Ogólnie: do weryfikacji dedukcyjnej fragmentów matematyki i informatyki

W szczególności:

- ▶ formalizacja metateorii języków programowania
- ▶ weryfikacja programów
- ▶ generowanie kodu (poprzez ekstrakcję z dowodu)
- ▶ współpraca z zewnętrznymi narzędziami (np. Why 3, Frama C)

PRZYKŁADOWE PROJEKTY ZREALIZOWANE W COQU

- ▶ CompCert (X. Leroy, INRIA) – certyfikowany kompilator języka C
- ▶ formalizacja semantyki (fragmentu) języka C++ (X. Leroy)
- ▶ weryfikacja Java Card
- ▶ formalizacja algebry relacji i DBMS (projekt Ynot)
- ▶ certyfikowane serwisy www (projekt Ynot)
- ▶ certyfikowany kompilator synchronicznego języka Lustre (M. Pouzet, INRIA)

CZEGO BĘDZIEMY SIĘ UCZYĆ

- ▶ jak modelować różne pojęcia i konstrukcje matematyczne/informatyczne w języku specyfikacji Coq (w teorii typów)
- ▶ jak programować w języku z typami zależnymi
- ▶ jak formalnie dowodzić twierdzeń
- ▶ jak automatyzować dowodzenie twierdzeń

PO CO CHCEMY SIĘ TEGO UCZYĆ

- ▶ formalna weryfikacja daje pewność poprawności dowodu, algorytmu itp. (przy założeniu adekwatnej reprezentacji problemu, poprawności implementacji itd...)
- ▶ formalna weryfikacja ujawnia luki w rozumowaniu i pomaga wykrywać błędy
- ▶ przez swoją formalność dowodzenie bywa bardzo uciążliwe, dlatego chcemy tę czynność automatyzować
- ▶ Coq wspiera tworzenie bezpiecznego, certyfikowanego oprogramowania
- ▶ budowanie bibliotek zweryfikowanych problemów
- ▶ POPLmark challenge (formalizacja metateorii języków programowania)

POTRZEBNE NARZĘDZIA

- ▶ definicje indukcyjne obiektów i własności
- ▶ dowód przez indukcję strukturalną; dowód przez indukcję po drzewie wyprowadzenia
- ▶ wnioskowanie w systemach dedukcji naturalnej w logikach intuicjonistycznej i klasycznej
- ▶ beztypowy rachunek lambda
- ▶ rachunek lambda z typami prostymi