

Ekstrakcja programów w Coq

1. WSTĘP

Coq jest interaktywnym systemem wspomagającym dowodzenie twierdzeń. Pozwala precyzyjnie formułować twierdzenia matematyczne i mechanicznie sprawdzać dowody ich poprawności, pomaga znaleźć dowody i ekstrahuje programy z formalnych dowodów ich specyfikacji. *Coq* działa w teorii rachunku konstrukcji induktywnych, który jest wariantem rachunku konstruktorów. *Coq* nie jest programem do automatycznego dowodzenia twierdzeń, ale zawiera pewne taktyki z zakresu automatycznego dowodzenia twierdzeń i procesy decyzyjne. *Coq* implementuje też zależny typowy język programowania.

Więcej do poczytania o *Coq* tutaj:

<http://coq.inria.fr>

Tam też do znalezienia oprogramowanie i tutoriale. Pomocne mogą okazać się też moje wykłady z logiki:

<http://www.math.us.edu.pl/~pgladki/teaching>

Celem niniejszego projektu jest zapoznanie się z systemem *Coq* i zobaczenie, jak można wykorzystać go do weryfikacji programów.

2. ZADANIE

Definiujemy typ

Inductive bintree : Set :=

leaf : nat -> bintree

— node : bintree -> bintree -> bintree.

”Obraz zwierciadlany” takiego drzewa otrzymujemy wymieniając lewe i prawe poddrzewa na każdym poziomie zagnieżdżenia (czyli rekursywnie).

(1) Sformułuj to pojęcie indukcyjnym predykatem

Inductive is_mirror : bintree -> bintree -> Prop

(2) Skonstruuj metodą ekstrakcji program zwracający obraz zwierciadłowy argumentu (czyli sformułuj odpowiednie twierdzenie w *Coq*’u, przeprowadź dowód tego twierdzenia i zastosuj komendę *Recursive Extraction* (uwaga na konstrukcyjność używanych typów!)).

(3) Zapisz rezultat (komendy *Cd* i *Extraction* odpowiednio sparametryzowane).

(4) Napisz program w *Coq*’u (*Fixpoint*) realizujący obraz zwierciadlany argumentu.

(5) Przeprowadź dowód jego poprawności.

(6) Wykonaj ekstrakcję tego programu i porównaj z rezultatem poprzedniej ekstrakcji. Co stwierdzasz? Czy potrafisz to sobie wytłumaczyć?