Hoare Advanced Homework Assistant

Aleksy Schubert[1] Tadeusz Sznuk[2]

[1]alx@mimuw.edu.pl

[2]tsznuk@mimuw.edu.pl

3 października 2013

Cele projektu

Cel

Przeniesienie nauczania semantyki na wydziale w XXI wiek.

- Na początek chcemy się skupić na nauczaniu logiki Hoare'a.
- Konkretnie zamierzamy stworzyć narzędzie wspierające ów proces.

Stan obecny

W tej chwili część zajęć z semantyki dotycząca logiki Hoare'a wygląda następująco

1. Na wykładzie przedstawione są teoretyczne podstawy.
2. W ramach ćwiczeń realizowana jest praca domowa, polegająca na uzupełnieniu zadanego programu o brakujące asercje.
3. Podobne zadanie pojawia się na egzaminie.

Zadania oddaje się w formie papierowej. Prowadzący sprawdzają zgodność rozwiązań z regułami logiki Hoare'a, co sprowadza się do zbadania prawdziwości pewnej liczby dość zawiłych formuł logicznych.

Problemy

- Sprawdzanie rozwiązania jest procesem niezwykle żmudnym.
- Problem ten dotyczy nie tylko prowadzących zajęcia - student nie może się łatwo przekonać, czy jego rozwiązanie jest poprawne (i dlaczego nie jest).
- Co gorsza, przy takim tworzeniu programu w logice Hoare'a łatwo się pomylić. Zmniejsza to znacznie szanse otrzymania poprawnego programu, co powoduje zwątpenie w sens stosowania logiki Hoare'a (czy nawet wszelkich metod formalnej weryfikacji).

Pomysł

Pomysł 1 - IDE

- Programów nie pisze się na kartkach.
- Programy pisze się w edytorach.

Pomysł 2 - Automatyczna weryfikacja

- Sprawdzanie prawdziwości formuł logicznych jest zwykle zadaniem żmudnym.
- A zatem należy do tego celu wykorzystać komputer.
- Ten pomysł nie jest nowy - istnieje już sporo pożytecznych narzędzi (SMT-solverów, np. Microsoft Z3).

Co mamy

- Prototyp narzędzia
- Spisane w formie stosownych dokumentów wymagania.
  - Odnośnie całego narzędzia.
  - Oraz proponowanych podprojektów.
- Dużo dobrych chęci?

Prototyp

[[Może się uruchomi]].

Zalety projektu

- Wynik prac będzie stosowany w praktyce przez przyszłe pokolenia studentów.
- Źródłem wymagań są prowadzący przedmiot oraz studenci.
  - Tych pierwszych można zwykle zastać na wydziale.
- Udział w realizacji projektu może się okazać pomocny przy nauce (oraz zaliczaniu egzaminu z) semantyki.
- Doświadczenie w tworzeniu projektów eclipsowych może być przydatne.
- Prace związane z projektem można kontynuować po zakończeniu ZPP (w ramach grantów czy prac magisterskich).

Proponowane projekty

- Dalszy rozwój prototypu, w szczególności interfejsu.
- Implementacja narzędzia (być może za pomocą innych technik, niż zastosowane w prototypie).
- Eclipsowe API do współpracy z SMT-solverami + prosty solver.
- Interpreter i debugger (zintegrowane z Eclipse).
- ...

Implementacja narzędzia

- W tej chwili mamy prototyp.
- Jego interfejs jest jednak raczej niedopracowany.
- Potrzeba dużo pracy, aby nadawał się do codziennego użytku na zajęciach.
- Platforma Eclipse oferuje ogrom możliwości.
  - Niestety, nierzadko są to możliwości stracenia czasu i/lub nerwów.
  - Być może zastosowanie innej technologii umożliwi sprawniejszą realizację projektu.

SMT solvery

- SMT-solver służy do sprawdzania spełnialności formuł logicznych (nad jakąś teorią, np. liczbami całkowitymi i tablicami).
- Istnieje sporo implementacji. Obecnie najbardziej obiecująco wygląda Microsoft Z3.
- Istnieje nawet standard do komunikacji z solverami (SMTLIBv2). Ale żaden solver go tak do końca nie implementuje.
- Potrzebna jest wtyczka eclipsowa, umożliwiająca innym projektom korzystanie z solverów (a użytkownikom - konfigurowanie zainstalowanych solverów).
- Dobrze byłoby mieć też własny solver.

Interpreter

- Aby uzupełnić dowód poprawności programu, dobrze jest zwyke zrozumieć, jak (czy?) ten program działa.
- Możliwość uruchomienia wybranego fragmentu kodu dla zadanych parametrów bywa w tym pomocna.
- Przydatne jest też podglądanie wartości zmiennych w różnych punktach, zatrzymywanie wykonania itp.
- Eclipse zawiera interfejs do odpluskwiania programów działających na JVM - może można z niego skorzystać?