COSC 3341 3.0 Introduction to Program Verification

Winter 2001

Brief overview

Every program implicitly asserts a theorem to the effect that if certain input conditions are met then the program will do what its specifications or documentation says it will. Making that theorem true is not merely a matter of luck or patient debugging; making a correct program can be greatly aided by a logical analysis of what it is supposed to do, and for small pieces of code a proof that the code works can be produced hand-in-hand with the construction of the code itself. Good programming style works in part because it makes the verification process easier and this in turn makes it easier to develop more complex algorithms from simple ones.

The course will provide an introduction to the basic concepts of formal verification methods. It will also include the use of simple tools to aid in verification.

Topics covered will include the following:

* The role of formal verification in the software life-cycle; verification versus testing and validation. Varieties of verification "scenarios".
* Introduction to propositional calculus; checking for tautologies and contradictions; annotating code with assertions.
* Verification scenarios for abstract datatypes; using induction.
* Symbolic execution; proving relative correctness for small code segments; establishing termination.

General information for Section M

Time: Monday, Wednesday and Friday, 12:30–13:30
Place: CCB 115
Instructor: Franck van Breugel
Office: CCB 348
Office hours: Monday, Wednesday and Friday, 13:30-14:30 or by appointment
Email: franck@ariel.cs.yorku.ca

General information for Section N

Time: Monday, 19:00–22:00
Place: CCB 115
Instructor: Rachel Jiang
Office: CCB 154
Office hours: Monday, 17:00-18:30 or by appointment
Email: rachel@cs.yorku.ca

Reference material

The text for the course is

* Peter H. Roosen-Runge. *Software Verification Tools*. 2000.

This text is available at the URL

http://www.cs.yorku.ca/course/3341/

The following book is suggested for further reading.

* Roland C. Backhouse. *Program Construction and Verification*. Prentice-Hall. 1986.

This book is on reserve in the Steacie library.

Evaluation

The student's performance in the course will be evaluated as a combination of a final exam (50%), a midterm (30%) and assignments (20%). More details are given below. There will be no supplemental examination for the course. Neither will students have the option of doing additional work to upgrade their mark.

Assignments: There will be two assignments. The assignments are given out on

1. March 12 and
2. April 16.

These assignments should be handed in within two weeks. No late assignments will be accepted. If a student cannot hand in the assignment in time for reasons beyond his/her control, the student should bring a documented note to the instructor. If accepted, the weight of the other assignment will be prorated accordingly. The assignments will be returned within two weeks after the due date. The assignments will be posted at the URL

http://www.cs.yorku.ca/course/3341/

Midterm: The midterm will be held on April 2. The midterm will be written in class. If a student misses the midterm for reasons beyond his/her control, the student has to bring a documented note to the instructor. If accepted, the weight of the final exam will be prorated accordingly.

Final exam: The final exam will be held in the examination period. It will be a two hour exam.

Additional information can be found at the URL

http://www.cs.yorku.ca/course/3341/