CAS 743 — Functional Programming

2006-01-09

Outline

Instructor: Dr. Wolfram **Kahl**
Department of Computing and Software
McMaster University

Room: ITB-245

E-Mail: kahl@mcmaster.ca

Course Pages: http://www.cas.mcmaster.ca/~kahl/FP/2006/

This is where you find further information, announcements, and useful links. Electronic versions of the assignment sheets will also be kept there.

It is the student's responsibility to be aware of the information in the course Web pages, and to check regularly for announcements.

Literature:

Useful Textbooks (choice of <u>one</u> is recommended):

Simon Thompson. *The Craft of Functional Programming.* Pearson — Addison Wesley. Second Edition, 1999. ISBN: 0-201-34275-8 (uses Haskell 98)

Richard Bird. *Introduction to Functional Programming using Haskell*. Prentice Hall Series in Computer Science. Prentice Hall Europe, 1998. ISBN: 0-13-484346-0 (appeared before Haskell 98, slightly more decoupled from language details)

Also recommended:

Hal Daume III: *Yet Another Haskell Tutorial*, http://www.isi.edu/~hdaume/htut/

Haskell Website: http://haskell.org/

Schedule:

Monday, Thursday, 8:30–10:00

Grading:

- 30% Assignments (details to be announced)
- 15% Presentation (details to be announced)
- 20% Midterm (details to be announced)
- 35% Final Exam

All examinations in this course will be **Closed Book**. That is, no written or printed material nor a calculator may be used during the examinations.

The instructor reserves the right to conduct any deferred exams orally.

Contents

The major topics are most likely to be presented in an interleaved manner that allows applying theory in practice with less delay.

- Practice of Pure Functional Programming (in Haskell)
  - Values, simple functions, primitive datatypes, lists, list comprehension
  - Common higher-order functions like fold, map, filter; repetition and iteration patterns
  - Function definitions using pattern matching, recursive functions over lists
  - Type classes and instances
  - Datatype definitions, abstract datatypes, the role of class instances in datatype abstraction
  - Computations and State in the purely functional setting: Monads, I/O
  - Advanced datatype abstraction using multiparameter classes; monad transformers; modular interpreters
  - Combinator libraries and embedded domain-specific languages; parsing combinators
- Foundations of Functional Programming:
  - Term rewriting systems, λ-calculus, general concepts of reduction systems, evaluation strategies
  - Type systems, parametric polymorphism, ad-hoc polymorphism, second-order polymorphism, Hindley-Milner typing
  - Category theory
- Impure Functional Programming
- Implementation of Functional Programming Languages

Official Statements

Academic Dishonesty

Academic dishonesty consists of misrepresentation by deception or by other fraudulent means and can result in serious consequences, e.g. the grade of zero on an assignment, loss of credit with a notation on the transcript (notation reads: "Grade of F assigned for academic dishonesty"), and/or suspension or expulsion from the university.

It is your responsibility to understand what constitutes academic dishonesty. For information on the various kinds of academic dishonesty please refer to the Academic Integrity Policy, specifically Appendix 3, located at http://www.mcmaster.ca/senate/academic/ac_integrity.htm.

The following illustrates only three forms of academic dishonesty:

(1) Plagiarism, e.g. the submission of work that is not one's own or for which other credit has been obtained.
(2) Improper collaboration in group work.
(3) Copying or using unauthorised aids in tests and examinations.

Discrimination

The Faculty of Engineering is concerned with ensuring an environment that is free of all adverse discrimination. If there is a problem that cannot be resolved by discussion among the persons concerned, individuals are reminded that they should contact the Department Chair, the Sexual Harassment Office or the Human Rights Consultant, as soon as possible.