Preliminary Version of 27 July

Course Information

Lectures

Tuesdays and Thursdays, 11:00-12:15, in Mechanical Engineering 220.

Instructor

Darko Stefanovic, office hours Tuesdays and Thursdays 10:00-11:00 in FEC 345C.

Teaching assistant

None

Course topics and format

The course is an informal introduction to the theory used to describe and define programming languages, and to guide their implementation. Our approach is type-based, in the spirit of our textbook, Pierce's *Types and Programming Languages* (TAPL). As a prelude, the course offers an overview of programming techniques and programming language features found in the purely functional programming language Haskell.

The course is intended for first-year graduate students, but advanced undergraduates are welcome as well. No specific courses are prerequisites, but programming experience and mathematical maturity are necessary. In particular, experience with functional programming (at the level of UNM CS357) is important. Facility with formal system manipulation is essential.

The course will provide students with the background they need for CS550 (offered in the spring).

The course consists of lectures, written assignments, extensive programming assignments, two mid-term examinations, and a final examination.

Assignments

There will be two *in-class* midterm exams (the first is likely to be given right before we start on TAPL, the second after we have completed discussion of the simply-typed lambda calculus, i.e., after Chapter 11). There will be a final exam covering the entire course. Several short written homework assignments may be given to consolidate lecture material; they may take the form of short algebraic proofs of program fragment equivalence, or consideration of small language extensions. Several programming assignments may be given: in the early part of the course the tasks will be drawn from the general domains of mathematics, science, and engineering, to practice programming skills; in the latter part of the course the tasks will correspond to implementation of programming language theory.

Textbooks

Benjamin C. Pierce, *Types and Programming Languages*, MIT Press, 2002, ISBN-10: 0262162091.

Miran Lipovača, *Learn You a Haskell for Great Good*, No Starch Press, 2011, ISBN-13: 978-1-59327-283-8.

Grading

You are expected to attend class regularly, *read the assigned reading before class*, and participate in class discussion. The grade will be determined as follows:

Homeworks 50%

Exams 50% (15% each midterm exam, 20% final)

Topics

1. functional programming and Haskell introduction
2. prelude types and classes
3. functions and list comprehensions; unit testing; literate programming; interactive programs
4. recursive and higher-order functions
5. declaring types and classes
6. lists in depth: map, filter, and their algebraic laws
7. lists in depth: foldr, scanr, and their algebraic laws
8. trees with folds, binary heap trees, rose trees
9. efficiency: accumulating parameters, tupling, fusion and deforestation
10. modules and abstract data types
11. lazy evaluation and infinite data structures; approximation ordering; cyclic structures; streams
12. monads
13. syntax
14. operational semantics
15. lambda calculus syntax and reduction
16. programming in the lambda calculus
17. combinators and combinator reduction

18. types
19. simply typed lambda calculus
20. simple extensions (ascription; let-bindings; records)
21. simple extensions (variants; recursion)
22. references
23. exceptions
24. subtyping
25. recursive types
26. type reconstruction
27. unification
28. universal polymorphism

Mailing list

A mailing list will be used for class discussion. It may also be used for administrative announcements. See `http://www.cs.unm.edu/cgi-bin/mailman/listinfo/cs558swf`.

Wiki

The permanent location for course material, such as assignments and shared files, is the course wiki, `https://digamma.cs.unm.edu/wiki/bin/view/UNMCS558Fall2011Web/WebHome`.

UNM statement of compliance with ADA

Qualified students with disabilities needing appropriate academic adjustments should contact the instructor as soon as possible to ensure their needs are met in a timely manner. Handouts are available in alternative accessible formats upon request.