Preliminary version of 20 January

Course Information

Meetings

Tuesdays and Thursdays, 2:00–3:15, in FEC345.

Instructor

Darko Stefanovic, office hours Tuesdays and Thursdays, 3:20–4:00 in ECE 236C.

Course topics and format

A seminar with readings and presentations on selected topics in typeful functional programming. Readings range from classics to current research, but emphasis is on tutorial articles on programming techniques, principally from the summer schools on advanced functional programming.

Reading List / Lecture Plan

This is a preliminary list of topics, which will be adjusted to reflect the interests of course participants. The pace will be adjusted as well; we may take more than one class period per paper if the discussion shows this is necessary. The sequence proposed below allows dropping the tail. I expect, however, that we will definitely cover topics 1–15. Any dropped topics will be covered in a seminar next year.

Presenter name shown in italics.

Week 1-2: Introduction
1 organizational; Haskell in brief [IFPH, Ch1-5; H9] Stefanovic
2 functional programming [H4] [T1] Stefanovic
3 trees with folds, binary heap trees, rose trees [IFPH, Ch6] Stefanovic

Week 3: Efficiency
4 lazy evaluation, accumulating parameters, tupling [IFPH, Ch7]
5 fusion and deforestation [IFPH, Ch7]

Week 4-6: Data structures
6 amortization; queues and splay heaps [O1]
7 red-black trees [O6]
8 more heap trees [O7]
9 tree traversal [O3]
10 tree access [H1]
Week 7: Types and classes
11 type classes; constructor classes [W4] [J3]
12 extensions: multi-parameter type classes; functional dependencies in type classes [J2]
13 examples: math education [F1]

Week 8-9: Combinators
14 intro to parsing combinators [F2] [H7]
15 pretty-printing combinators [H5]

Week 10-11: Monads
16 monads, combining monads, monadic IO [W1] [IFPH, Ch10]
17 monads and other effects [PJ1]

Week 12-13: More combinators
18 monadic parsing combinators [H8] [L3]
19 combinator languages [S1]
20 pretty printing [W3]
21 beyond monadic combinator parsing [S2]

Week 14: Arrows
22 arrows [H3]

Week 15: Functional reactive programming
23 Yampa [H10]

Week 16: Various
24 types and matrices [O5]
25 origami programming [G2]

Sources: Papers

F1 Jeroen Fokker. Explaining algebraic theory with functional programs.
F2 Jeroen Fokker. Functional parsers.
H5 John Hughes. The Design of a Pretty-Printing Library. AFP.
H8 Graham Hutton and Erik Meijer. Monadic Parsing Combinators.


L3 Daan Leijen and Erik Meijer. Direct Style Monadic Parser Combinators For The Real World.

O1 Chris Okasaki. Functional Data Structures, AFP.


PJ1 Simon Peyton Jones. Tackling the Awkward Squad.

S1 S. Doaitse Swierstra, Pablo R. Azero Alcocer, and João Saraiva. Designing and Implementing Combinator languages. AFP.

S2 S. Doaitse Swierstra and Luc Duponcheel. Deterministic, Error-Correcting Parsers. AFP.


W1 Philip Wadler. Monads for Functional Programming. AFP.


Sources: Books

Presentations

Presentations should be around 40 minutes long, to allow time for discussion. Prepared slides are recommended but not required; you may use a few slides from the author’s or someone else’s presentation, with acknowledgement.

Most of the papers we will read describe implementations of particular programs or libraries. A good presentation will develop new examples and (update and) test the code.

There will be a mechanism for peer review of presentation quality, and it will be part of the course grade.

Grading

Presentation preparedness and quality, in-depth analysis of material, participation in the discussion.

Mailing list

A mailing list will be used for class discussion. It may also be used for administrative announcements.

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.