Advanced Functional Programming

2D1456, Spring 2007, January-March, D-level Graduate Course
School of Computer Science and Communication, KTH

http://www.csc.kth.se/utbildning/kth/kurser/2D1456/avfunk07/

Course Start

First lecture: room D32 KTH campus at 13.15 17/1 2007.
Second lecture: room D33 15.15 18/1 2007.

Teacher

Johan Glimming <glimming@kth.se>
Tel. 08-7909289, mailbox 4th Floor, Lindstedtsvägen 3, Osquars backe 2, KTH

Student's Office

http://www.csc.kth.se/student/studentexpedition/
Lindstedtsvägen 3, 2nd Floor (ground level), Osquars backe 2, KTH

Learning Goals

The goals of the course is to give the students

- experience in using advanced concepts in modern strongly typed functional programming languages
- experience from implementation of (purely) functional algorithms and data structures, and the analysis of these
- understanding of some basic program transformations and optimisation methods for functional programs
- basic understanding of the theoretical foundations (lambda calculus/type theory) for functional languages

for the students to

- be able to solve practical problems using functional languages, particularly algorithmic problems
- be able to explain the relationship between functional languages and e.g. imperative and object-oriented languages
- solve basic mathematical problems in the surrounding theory and understand the relevance of this theory for the construction of new languages.

Registration

Using your CSC computer account, type the following two commands:

```
res checkin avfunk07
res course join avfunk07
```

Course Books

The following two titles can be purchased e.g. at Kårbokhandeln, KTH Campus (on the back of KTHB Library):

- Richard S. Bird, *Introduction to Functional Programming using Haskell*, Second Edition, 1998.

- Benjamin C. Pierce, *Types and Programming Languages*, MIT Press, 2002. (Selected chapters: 5, 9, 11, 22, 23)

It is important that students have access to both books. The first book is used primarily for lecture 1-7 and the second book for the remaining lectures 8-14 but it is also needed for several mini projects. KTHB Library can help.

Examination and Grade

The course consists of two parts: 2 credits for the mini-project (including your own report and opposition) and 2 credits for completed homework assignments. Each part is awarded a separate grade 3, 4, or 5, and your final grade is computed based on these two grades. The only mandatory part of the course is presence on those lectures involving delivery of homework (see homework deadlines).

Lecture Plan

The lectures have the following preliminary contents and are given in English:

Part 1 – Practical Programming
L1 Introduction
L2 Basic functions and datatypes
L3 Purely functional algorithms and data structures
L4 Purely functional algorithms and data structures
L5 Monads
L6 Parser monads
L7 Monadic interpreters and monad transformers

Part 2 – Theory
L8 Untyped lambda calculus
L9 Simply typed lambda calculus
L10 Polymorphic lambda calculus
L11 More polymorphism, Type inference
L12 Type inference
L13 Implementation techniques
L14 Presentation/Opposition of mini-projects

Labs / Mini-Projects

As part of the course, students are to complete mini-projects, i.e. advanced programming projects carried out in Haskell. In these, one topic (or more) of the course is to be handled in practical programming. Students are to present their results on the last lecture and in a written report. Such projects can be chosen from a predetermined list of specifications, but students are also welcome to propose different specifications relevant to the course contents (e.g. an interpreter for a domain-specific language). A lab session is scheduled every week where the teacher can offer guidance and help.

Project specifications are listed on the course web page (see above). Allocation is on first-come first-served basis.

Homework

For most weeks, there will be a set of assignments, which are to be completed individually. These assignments can be downloaded from the course web page.

Assignments are used during lectures/presented/discussed. The deadline for homework is strict.

Homework Deadlines:

H1	Wednesday January 24	(bring to lecture)
H2	Wednesday January 31	(bring to lecture)
H3	Wednesday February 7	(bring to lecture)
H4	Wednesday February 14	(bring to lecture)
H5	Friday March 9 – final exam	(electronically or using teacher's mailbox)

Mini-project Deadlines:

Own project report / project finished	**Thursday February 22**
Opposition of other project	**Thursday March 1**

Schedule

14 lectures + 6 labs

Week 3, 2007					Type	Place
		Wed	17/1	13:15-15:00	Lecture	D32
		Thur	18 /1	13:15-15:00	Lecture	D33
				15:15-17:00	Lab	5V1Grå
Week 4, 2007						
		Wed	24/1	13:15-15:00	Lecture	K53
		Thur	25/1	13:15-15:00	Lecture	M31
				15:15-17:00	Lab	5V1Grå
Week 5, 2007						
		Wed	31/1	13:15-15:00	Lecture	E33
		Thur	1/2	13:15-15:00	Lecture	E3
				15:15-17:00	Lab	5V1Grå
Week 6, 2007						
		Wed	7/2	13:15-15:00	Lecture	V23
		Thur	8/2	13:15-15:00	Lecture	E3
				15:15-17:00	Lab	5V1Grå
Week 7, 2007						
		Wed	14/2	13:15-15:00	Lecture	D31
		Thur	15/2	13:15-15:00	Lecture	D33
				15:15-17:00	Lab	5V1Grå
Week 8, 2007						
		Wed	21/2	13:15-15:00	Lecture	D35
		Thur	22/2	13:15-15:00	Lecture	D34
				15:15-17:00	Lab	5V1Grå
Week 9, 2007						
		Wed	28/2	13:15-15:00	Lecture	D33
		Thur	1/3	13:15-15:00	Lecture	D34

Reading instructions are published on-line for each lecture.

Course Evaluation and Summary of Course Analysis

At the end of the course, students are invited to evaluate the course using a web-based feedback form. The previous year's evaluation indicated that the course was quite demanding. However, most students that attended lectures, completed course homework, and came to scheduled labs also passed the course. Many projects were completed well and/or with extensions, which resulted in high grades.

Code of Honour

All students must read the following statement:
http://www.csc.kth.se/utbildning/hederskodex/

Mini-Project Grading

When submitting your project (deadline: February 22) you are required to provide:

1. Clearly written and commented Haskell (GHC) source code complying to the style guidelines published on the course web page.
2. A statement of who has done what in the project (if more than one person worked on it).
3. Test cases: a separate file (e.g. Test-ModuleA.hs) with Haskell code for the test, and a comment showing the expected (correct) output
4. A *project report* which describes the implementation and your approach, and discusses also the project in relation to relevant topics covered by the course. The report should specifically contain the following sections:
    a. *Discussion*, in which you discuss problematic issues that arose and questions posed in this project specification. Also try to be problem-oriented and discuss interesting problems you identified or would have liked to study further. Discuss if functional programming was useful or not.
    b. *Efficiency*, in which you discuss the space and time efficiency of critical functions/algorithms used in the project. Convince the reader that your approach is feasible in terms of complexity.
    c. *Overview*, which describes your chosen implementation.
    d. *Test Cases*, which lists the tests you have performed and their results.

Literature required for the mini-project is on-line, accessible via KTHB Library, or made available by the teacher (see the literature sections on the course web page).

Before the presentation of the projects (on the last lecture March 1) each project also prepares a set of questions/remarks based on studying a different group's project and project report. Instructions for this opposition are given later in the course.

Project grading takes into account the inherent difficulty of the mini-project (typically stated in the specification), the quality of your report (particularly the discussion), and on how well contents of the course have been used in the solution, e.g. efficient functions, monads, higher-order functions, suitable datatypes, and such like. In addition, oppositions and project presentations are considered for the grade.

Attendance on lab sessions is highly recommended for a successful project.