Spring 2007

CS 776: Functional Programming

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
https://corescholar.libraries.wright.edu/cecs_syllabi/400

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
CS 776 Functional Programming

- Instructor: T. K. Prasad
- Phone No.: (937)-775-5109
- Email: t.k.prasad@wright.edu
- Home Page: http://www.cs.wright.edu/~tkprasad
- Class Hrs: TTh, 12:20 - 1:35pm, 309 Oelman
- Office Hrs: TTh, 1:40pm-2:10pm, 395 JC (or by appointment)

Course Description

This course will discuss important concepts of functional programming such as recursive definitions, higher-order functions, type inference, polymorphism, abstract data types, modules etc. The programming exercises will illustrate the utility of list-processing, pattern matching, abstraction of data/control, strong typing, and parameterized modules (functors). We also study the mathematical reasoning involved in the design of functional programs and techniques for proving properties about functions so defined.

The programming assignments will be coded in SML '97 (Standard ML of New Jersey). and Haskell.

Prerequisites

- CS 480/680 Comparative Languages.

Course Texts


References

Course Load

The course load includes a mixture of homeworks and programming assignments worth 50 points, a midterm worth 20 points and a final worth 30 points. Exams are typically open book.

Grading

The letter grades will be assigned using the following scale: A[90-100], B[80-90], C[70-80], D[60-70), and F[0-60). However, I reserve the right to adjust the scale somewhat to utilize the gaps in the distribution.

Attendance Policy

All registered students are expected to attend all lectures. In case a student is absent from a lecture due to unavoidable circumstances, the student is still responsible for the material covered in the class, as it is typically available from the course web-page well in advance. Furthermore, the student is expected to find out about in-class announcements from their colleagues/instructor.

Class Schedule and Syllabus

<table>
<thead>
<tr>
<th>No.</th>
<th>Topic</th>
<th>Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>Functional Programming Basics</td>
<td>Chap. 1 (LP) Chap. 1 (JU)</td>
</tr>
<tr>
<td>Class 2</td>
<td>Higher-order functions</td>
<td>Chap. 2, 5 (LP) Chap. 2, 5 (JU)</td>
</tr>
<tr>
<td>Class 3</td>
<td>Type inference</td>
<td>Chap. 2, 3 (LP)</td>
</tr>
<tr>
<td>Class 4</td>
<td>Polymorphic Type System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Fixed Points)</td>
<td></td>
</tr>
<tr>
<td>Class 5</td>
<td>Programming with lists: Pattern matching</td>
<td>Chap. 3 (LP) Chap. 3-5 (JU)</td>
</tr>
<tr>
<td>Class 6</td>
<td>SML-97 Specifics</td>
<td></td>
</tr>
<tr>
<td>Class 7</td>
<td>Introduction to Haskell</td>
<td></td>
</tr>
<tr>
<td>Class 8</td>
<td>Fold operations: foldr, foldl</td>
<td>Chap. 5 (LP) Chap. 5 (JU)</td>
</tr>
<tr>
<td>Class 9</td>
<td>Midterm Exam (April 24)</td>
<td></td>
</tr>
<tr>
<td>Class 10</td>
<td>Types: Concrete and Abstract</td>
<td>Chap 6 (JU)</td>
</tr>
<tr>
<td>Class 11</td>
<td>Examples</td>
<td>Chap 7 (LP)</td>
</tr>
</tbody>
</table>
## Assignments (Spring 2006)

- Assignment 1 (asg1.html)
- Assignment 2 (asg2.html)

## Examinations (Spring 2004)

- Midterm
- Final

---

*T. K. Prasad (03/28/2007)*