Spring 2013

CS 7120: Functional and Logic Programming

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

CS 7120 Functional and Logic Programming

- **Instructor** : T. K. Prasad
- **Phone No.** : (937)-775-5109
- **Email** : t.k.prasad@wright.edu
- **Home Page** : http://knoesis.wright.edu/tkprasad

- **Quarter** : Spring, 2013.
- **Class Hrs** : TTh, 5-6:20pm, **129 Med Sci**
- **Office Hrs** : TTh, 3:30-4:30pm, 395 Joshi (or by appointment)

Course Description

This course will discuss important concepts and language features to support (i) functional programming and (ii) logic programming. Specifically:
(i) The first half of the course will cover functional programming techniques and constructs such as recursive definitions, higher-order functions, type inference, polymorphism, abstract data types, and modules. The programming exercises will illustrate the utility of list-processing, pattern matching, abstraction of data/control, strong typing, and parameterized modules (functors). We will also study the mathematical reasoning (induction) involved in the design of functional programs and for proving properties about functions so defined. The programming assignments will be coded in SML '97 (Standard ML of New Jersey) and Haskell.
(ii) The second half of the course will cover logic programming paradigm and Prolog. We discuss the syntax and the semantics of Prolog, the workings of a Prolog interpreter, and various applications of Prolog. We consider the use of declarative programming and pattern matching for database querying, parsing, meta-programming, and problem solving in AI.The programming assignments will be coded in XSB and/or SWI_Prolog.

Prerequisites

- CS 3180/5180 Comparative Languages.

Course Texts

1. ML for the working programmer (2nd Ed.), L. C. PAULSON, Cambridge University Press, 1996. ISBN 0-521-56543-X.
2. Ivan Bratko, Prolog Programming for Artificial Intelligence (Fourth Edition). Addison-Wesley Publ. Co., 2012. ISBN-13: 9780321417466.
3. **PLUS: Online course material.**

Functional Programming References

1. Elements of ML Programming (ML 97 Ed.), J. D. ULLMAN, Prentice Hall, 1998. ISBN 0-13-790387-1.
2. Introduction to Functional Programming in Haskell (2nd Ed.), R. S BIRD, International Series in Computer Science. Prentice Hall, 1998. ISBN: 0-13-484346-0
3. Programming in Standard ML , Robert Harper, Carnegie Mellon University, 2011.
4. A Gentle Introduction to Haskell , Paul Hudak, John Peterson, Joseph Fasel, 2000.
5. Why Functional Programming Matters, John Hughes.
6. The Semantic Elegance of Applicative Languages, David Turner, Proceedings of the 1981 conference on Functional programming languages and computer architecture, pp. 85-92, 1981.
7. The Standard ML Basis Library, Edited by Emden R. Gansner, John H. Reppy, Cambridge University Press, 2004.

Logic Programming References

1. David S. Warren, Programming in Tabled Prolog. (Online)
2. W. F. Clocksin and C.S.Mellish, Programming in Prolog (Fourth Edition), Springer-Verlag, 2000.
3. Ulf Nilsson and Jan Maluszynki, Logic Programming and Prolog (Second Ediiton), 2000.

Course Load

The course load includes a mixture of homeworks and programming assignments worth 40 points, a midterm worth 30 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 or access them on Pilot in a timely manner. In any case, the students are 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

No.	Topic	Readings

Class 1	Functional Programming Basics; Higher-order functions	Chap. 1,2,5 (LP) Chap. 1,2,5 (JU)
Class 2	Type Inference; Polymorphic Type System	Chap. 2, 3 (LP) (Fixed Points)
Class 3	Programming with lists: Pattern matching	Chap. 3 (LP) Chap. 3-5 (JU)
Class 4	SML-97 Specifics	(JU)
Class 5	Introduction to Haskell	Adv Haskell (1) (2)
Class 6	Fold operations : `foldr, foldl`	Chap. 5 (LP) Chap. 5 (JU)
Class 7	Types : Concrete and Abstract	Chap 6 (JU) Chap 7 (LP)
Class 8	Modules	Chap 7 (LP) Chap. 8 (JU)
Class 9	Recursion and Induction	Chap. 6 (LP)
Class 10	Examples (Combinatorial Functions)	
Class 11	Efficiency; Streams;Records; Exceptions; References;	Chap 5, 8 (LP) Chap. 7 (JU)
Class 12	**MIDTERM EXAM (February 14)**	
Class 13	Lambda Calculus Optional (1) (2)	
Class 14	Logic Programming Paradigm	
Class 15	Prolog and Unification	
Class 16	Meaning of Prolog Programs	
Class 17	List Processing; Operators	
Class 18	Arithmetic; Structures	
Class 19	Controlling Backtracking	
Class 20	Negation as Failure and Built-in Predicates	
Class 21	Definite Clause Grammars	
Class 22	Meta-Programming/Interpreters	
Class 23	Constraint Logic Programming	Practical Applications
Class 24	Logic and Models : Semantics of Prolog Programs	
Class 25	(cont'd)	
Class 26	Query Evaluation Strategies; Efficiency	
	Final Exam (April 23 : 5:45pm-7:45pm)	

Assignments (`Spring 2013`)

- Assignment 1 (asg-FP.html)
- Assignment 2 (asg-LP.pdf)

Old Examinations

- FP-Midterm
- FP-Final

- LP-Midterm
- LP-Final