CSCI 2520: Dependently Typed Functional Programming

Bowdoin College

Spring 2024

Online Presence

web: bowdoin-csci-2520.github.io

slack: bowdoin-csci-2520.slack.com

Meetings

Mondays, Wednesdays, Fridays 9:05–10:00 in Searles 128

Instructor

Ed Morehouse (e.morehouse@bowdoin.edu) Searles 202
Office hours: Mondays 2:00–3:00PM, Tuesdays 3:00–4:00PM, Wednesdays 10:00–11:00AM and 2:00–3:00PM, and by appointment.

Course Description

An important aspect of the expressiveness of a programming language concerns what kinds of things it can manipulate as data. This course introduces programming in a dependently typed functional language. In such a setting both functions and types are ordinary data. This enables features such as higher-order and generic functions, as well as data whose type is computed from other data. One benefit of working in this setting is that properties of programs can be both expressed and proved within the language itself, providing much stronger guarantees of correctness than is possible using testing. In this course we will study dependently typed functional programming using the language Idris (www.idris-lang.org).

Learning Goals

Upon successful completion of this course students should:

- understand the concepts of simple, parameterized, and indexed inductive types,
- understand the concepts of structurally recursive, higher-order, and dependent functions,
- be familiar with algebraic interfaces used in functional programming, including monoids, functors, and monads,
- be familiar with the Curry-Howard correspondence, and its use to state and prove properties of programs.

Course Structure

The main components of the course are as follows.

lecture: There are two weekly lectures, which introduce new concepts together with examples of their application.

lab: There is a weekly lab session, in which students complete tasks designed to build experience and competence with the concepts presented in the lectures.

homework: Approximately every other week there is a homework assignment consisting of a number of short programming exercises intended to build further competence and provide a basis for assessing student progress.

project: There is a term project, consisting of a larger and less precisely specified programming task, allowing students to demonstrate creative problem-solving in addition to programming skills.

exam: There are two exams to assess students' mastery of course material.

Student Evaluation

Course grades are determined as follows:

Homework Problem Sets	40%
Programming Project	20%
Exams	40%

You have a budget of 5 *late days* that you may use as you like for course assignments (subject to college restrictions regarding reading period and coursework submission deadlines).

Schedule

A tentative schedule of topics follows.

week	beginning	topics
1	2024-01-22	introduction to Idris
2	2024-01-29	inductive types and recursive functions
3	2024-02-05	parameterized types and generic functions
4	2024-02-12	function literals and higher-order functions
5	2024-02-19	programming interfaces
6	2024-02-26	monadic I/O
7	2024-03-04	review and first exam
8	2024-03-11	SPRING BREAK
9	2024-03-18	SPRING BREAK
10	2024-03-25	totality for data and codata
11	2024-04-01	algebraic interfaces
12	2024-04-08	indexed types and dependent functions
13	2024-04-15	propositions as types and first-order logic
14	2024-04-22	inductive equality types
15	2024-04-29	record types and automation
16	2024-05-06	review and second exam

Academic Integrity

Collaboration and learning from one another are encouraged, while copying answers and cheating are forbidden. You are expected to be able to distinguish the two. If you are contemplating an action, and you're not sure into which category it falls, you should consider whether what you intend to submit for evaluation is the product of your own efforts and represents your own understanding of the concepts involved. If it is/does not, then you should not submit the work as your own.

The Computer Science Department maintains a Collaboration Policy. It applies to this course as follows.

course component	collaboration level
labs	0
homeworks	1
project	2
exams	3

Moreover, Bowdoin College imposes an Academic Honor Code, which you are expected to abide by in all of your courses, including this one.

Academic Accommodations

Your instructor is committed to fostering an accessible and inclusive learning environment where all students feel welcome, comfortable, and treated fairly. If you have any concerns or suggestions for improvement, or would like to request an individual accommodation, please let me know.