HW 2 – Algebraic Datatypes

CS 421 – Summer 2013

Revision 1.0

Assigned June 7, 2013
Due June 14, 2013, 11:59 PM
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Your answers to the following questions are to be submitted electronically via the `handin` script as though an MP.

3 Objectives and Background

The purpose of this HW is to test your understanding of

- how to use OCaml variant types (a.k.a algebraic data types) to model specific examples

Another purpose of HWs is to provide you with experience answering non-programming written questions of the kind you may experience on the midterms and final.

4 What to `handin`

Answer each problem below, save your work as a PDF (either scanned if handwritten or converted from a program), and hand in the PDF as `hw2.pdf`.

5 Problem

(15 pts) A *red-black tree* is a kind of tree sometimes used to organize numerical data. It has two types of nodes, black nodes and red nodes. Red nodes always have one piece of data and two children, each of which is a black node. Black nodes may have either: 1) one piece of data and two children that are red nodes; 2) one piece of data and two children that are black nodes; or 3) no data and no children (i.e., a leaf node). (This isn't a 100% accurate description of red-black trees, but assume that it is for this exercise.) Write a pair of mutually recursive OCaml datatypes that represent red nodes and black nodes in red-black trees. For full credit, the "data" should be able to have any type, that is, your type should be polymorphic in the kind of data stored in the tree.