Data Driven Recursion

9-13-2004
Opening Discussion

- What did we talk about last class?
- Most people like the pace. A few wanted faster, a few wanted slower.
Let's look at code that I've written to do complex numbers.

Note that we have a “constructor” to build the abstraction, then other functions to do things with it.
Flat Recursion

- Functions that only play with the top level of a list are sometimes called flat. Most of what we have written has been of this type. What is the exception?
- We can write some others.
  - Merge – merges two sorted lists into a single list.
  - Remove – removes all instances of an item from a list.
- Let's take a second to look a bit closer at how recursion works by tracing one of these functions.
Deep Recursion

- Functions that not only recurse over the elements of a list, but also into the elements of a list are said to use deep recursion.
- We have done one of these, deep-count. Let's write two others.
  - Remove-all – removes all occurrences of an item for a list and all its sublists.
  - Reverse-all – reverses the list and all the sublists in it to any depth.
If we spend the second half of the semester on mostly ML, how many people would buy an optional textbook on it for ~$50? I've been reading it and it is a nice book in many ways.

Assignment #2 has been posted. It's due Friday so look through it and see if you have questions.