Recursion

9-3-2004

Opening Discussion

- What did we talk about last class?
- Would anyone be willing to show us the use of some Scheme code that uses things we talked about last time?
- Minute Essays
  - Single quote does seem to go to a white space or the closing parenthesis.
  - Memory usage in Scheme and other functional languages.
  - Studying ActionScript 2.0 can't happen for me this semester.

Recursion

- Recursive functions are functions that call themselves. We can use these functions in most language, but many people shy away from them, even when they are the easiest way to do things. Because of the lack of side effects, recursion is the primary way we get anything to happen multiple times in Scheme and other functional languages.
- Instead of a loop variable that is altered, we have a function that calls itself with an altered argument and the argument acts like the loop variable

More Recursion

- Any recursive function must have a conditional to terminate the recursion. There must be some situation that prevents the function from calling itself.
- In Scheme we can write recursive functions that play with lists when we don't know exact positions of things. Let's write some example functions that do this.

Some Possible Functions

- length – number of elements in a list.
- deep-count – How many atoms are in the list and its sublists.
- number-list – Make a list that has the numbers from 1 to n.
- bound-number-list – Make a list from m to n.
- member? - Is the item in the list?
- append – Append two lists together.
- map-list – Apply a function to all the elements of a list and return a new list with the results.
  - This one is more complex. It shows an example of passing a function as an argument.

Tracing and Debugging

- Obviously, a challenging aspect in any language is how to trace through code and debug problems.
- The most straightforward way to do this is with print statements. In Scheme, the display function does this, but only displays one item.
- Your book gives a simple example of enhancing this to writeln. It also shows nice entering and leaving functions.

Minute Essay

- Write a function which, given a list, will return a list that contains every other element of the list.
- How comfortable are you with recursion? It can be a tough topic, but you can take comfort that we have at least two more chapters on it.