Sets and Relations

10-1-2004

Opening Discussion

- What did we talk about last class?
- I asked how you might be able to use the fact that functions are first class entities in Scheme to build a objects like in an object-oriented language. The answer here is that you put

Finishing off Abstracting Functions

- I spent too much time talking about the things we can do with functions last time and didn't have time to go into abstracting them so lets do that now.
- The basic idea is that we can create a function that can be used to create any other flat recursive function. The same is true for deep recursion.
- Let's look at the example for flat recursion and then use it to generate some functions.

Quantifiers

- Before we get into sets lets look at some methods that can be helpful in dealing with sets, the quantifiers both, neither, and at-least-one.
- These are curried functions that take a predicate to check elements with and returns a predicate that determines if the predicate is true for the proper number of those arguments.

Sets

- A set in mathematics is a collection of elements where an element is either in the set or not. Duplication is not an issue. In math sets are typically written with curly braces.
- We will define our sets be defining an empty set and the following operations:
  - Adjoin – returns a set where an element is added
  - Pick – returns an element of a set
  - Residue – returns a set minus an element
  - Empty-set? and Set? - simple predicates

Building Logic Operations

- We can play with sets in Scheme in a way very similar to what we do in abstract math. We do this by extending neither, both, and at-least-one to work on sets.
- These become the operations, none, for-all, and there-exists. It turns out that if we write none, the others can be easily built with compose.

Operations on Sets

- There are many operations that we could write dealing with sets. We want to try to build up a few today in class.
  - Set-equal? - test equality of sets.
  - Element – tests if an element is in a set
  - Contains – same as element, but opposite order of arguments
  - Superset and Subset
  - Cardinal – number of elements in the set
  - Intersection, union, and difference

Minute Essay

- The code we wrote today makes serious use of the ability to use functions as first class objects. What did you think about the style of the code?
- You really should read chapter 8 to get a good feel for how these things work. It's an extremely different mindset than how you are used to programming.