Using Continuations

11-29-2004
Opening Discussion

- What did we start talking about on the Monday before Thanksgiving (that's the day everyone was very wet)?
- Do you have any questions about the assignments?
Terms from Last Time

- Context – This is where a program is in its execution when a certain statement is reached.
- Escape routine – This would be a function call that terminates execution.
- Continuation – This is a control structure where the program remembers the context of a call, then can later execute an escape routine to return to that context.
Continuations in Scheme

- To use a continuation in Scheme we use the function `call-with-current-continuation`. We pass this function one argument, which is a receiver function. That function should take one argument which will be the continuation.
- At any point in the receiver we can call the continuation with a value and the program will jump back to the context of `call-with-current-continuation` and substitute that value.
- Note that the receiver should be curried.
Continuations as Break-outs

- One use of a continuation is to short-circuit some computation. In this regard it is similar to using a break statement in a loop in C.
- This can make things more efficient without significantly complicating the code.
- This use of a continuation always calls the continuation or throw it away.
Loops with Continuations

- We can also do loops with continuations if the continuation “escapes” from where it was created. The logic for this becomes significantly more complex.
- This general idea of having a continuation escape from where it was created can make continuations expensive because the call stack must be saved in order to preserve the continuation.
Co-routines

- Another construct that we don't see much in modern programming languages is the co-routine. This is a set of functions that can transfer control to one another without actually terminating.

- For example, two functions A and B can be set up so that we start A, then A starts B. B will then resume A where it had left off and later A will resume B again. The two functions continue to trade off control.
Co-routines with Continuations

- This type of programming structure can be implemented nicely with continuations. When A calls B it passes a continuation into B that B can use to resume A. Similarly, when B resumes A, it provides a continuation that A can later use to resume B.
Now that you know what continuations can be used for, do you think you would ever use them if your language provided them?

Let's discuss due dates.