Massachusetts Institute of Technology  
Department of Electrical Engineering and Computer Science  
6.945/6.905 Spring 2019  
Problem Set 7  

Issued: Wed. 3 April 2019  
Due: Fri. 19 April 2019  

Reading:  
SDF Chapter 5 -- Evaluation: Sections 5.3 and 5.4  
Some, perhaps useful background material  
SICP, From Chapter 4: section 4.3; (pp. 412--437)  

Heavy Evaluator Hacking  

In this problem set we build interpreters in a different direction. We start with the essential EVAL/APPLY interpreter, written as an analyzer of the syntax into a compiler of compositions of execution procedures -- a small combinator language. We will warm up by making modifications to this evaluator.  

Next, we will change the evaluator to include AMB expressions. To add AMB, the execution procedures will all have a different shape: in addition to the environment, each will take two "continuation procedures" SUCCEED and FAIL. In general, when a computation comes up with a value it will invoke SUCCEED with the proposed value and a complaint department which, if invoked, will try to produce an alternate value. If a computation cannot come up with a value, it will invoke the complaint department passed to it in the FAIL continuation.  

An important lesson to be learned here is how to use continuation procedures to partially escape the expression structure of the language. By construction, a functional expression has a unique value. However, in the AMB system an expression may be ambiguous as to its value... Think about how we arrange that to make sense!  

Separating Syntactic Analysis from Execution  
(Compiling to Execution Procedures)  

You should read Section 5.3 of SDF (and perhaps SICP section 4.1.7 for more background) carefully here. When you load "load-analyze.scm" you will get the evaluator described in SDF (similar to the one described in SICP, but generalized for extensibility).
To Do
----------

Exercise 5.12: Implementing n-ary procedures p.234
Exercise 5.13: Simplifying debugging p.235
Exercise 5.14: Constant folding p.235
Exercise 5.16: Compiling formal-parameter declarations p.236

AMB and Nondeterministic Programming

Now comes the real fun part of this problem set! Please read section 5.4 of SDF (and perhaps section 4.3 of SICP) carefully before starting this part. This interpreter requires a change in the interface structure of the execution procedures that code compiles into, so it is quite different. Of course, our system differs from the one in SICP in that it is implemented with generic extension capability. The loader for the interpreter extended for AMB is "load-amb.scm".

To Do
----------

Exercise 5.17: A puzzle p.243
Exercise 5.19: Assignment p.244
Exercise 5.20: Choice ordering p.245