CS 565 Out: February 13, 1997
Jens Palsberg Due: February 20, 1997

Homework 4:

Enhancing an Interpreter

For this problem, you will modify the interpreter discussed in class. The interpreter is in `interpreter.scm` on the CS 565 homepage. It uses the scanner and parser utilities found in `sllgen.scm`. You will find documentation (old lecture notes by Professor Wand) of these utilities in the system notes to lecture 2. Please submit just one file containing all the extensions defined in the following items.

1. Add to the language the following construction:

⟨expression⟩ ::= `min` ⟨expression⟩ ⟨expression⟩

Both expressions should be evaluated. Their results are intended to be integers. The whole expression returns the smallest of the two integers.

2. Add to the language the following construction

⟨expression⟩ ::= `do` ⟨expression⟩ `provided` ⟨expression⟩

If the second expression gives a "true value", then the first expression is evaluated and the result of that is the result of the whole expression. If the second expression does not give a "true value", then the value -1 is the result of the whole expression.

3. Add to the language the following two constructions

⟨expression⟩ ::= `freeze` ⟨expression⟩

⟨expression⟩ ::= `force` ⟨expression⟩

The idea is that `freeze` e does *not* not evaluate e right away. Instead, a suspended expression is produced (usually called "thunk"). If `force` is applied to a thunk, then the suspended expression is evaluated *in its static environment*, and the result is returned as the result of the `force` expression.