Project 3: Programming in ML, Scheme, and the λ-calculus — due Friday 4 May

Total number of points available on this project is 200. Full credit is equivalent to 0 points. In other words, this project is entirely optional.

In Project 1 you implemented an interpreter for pure λ-calculus in ML.

Tasks:

1. Translate (compile) a subset of Scheme into the λ-calculus, and evaluate using the reducer from Project 1 (which you may modify if you find it convenient to do so).

   This subset should include at the very least the following:

   - **`(define ...)`**
   - **`(lambda ...)`**
   - **`(let ...)`**
   - **`(letrec ...)`**
   - **`(+ ...)`**
   - **`(- ...)`**
   - **`(eq?  ...)`**
   - **`(cons ...)`**
   - **`(car ...)`**
   - **`(cdr ...)`**
   - **`(list ...)`**

2. Test and debug on various Scheme programs you wrote in CS257, suitably modified to use only the Scheme subset you implement here.

3. Report your observations about the speed of various phases, ease of implementation, etc.