Advanced Programming Languages (COP 4930/CIS 6930) [Spring 2015]
Assignment VII

Due Date: Wednesday 4/22/15 at 5pm (in hardcopy)

Assignment Description

Do the following by yourself.

1. Prove the type-substitution lemma for a polymorphic but otherwise simply typed lambda calculus having base type nat. The only expressions in this language are function applications and abstractions (with types annotated on abstractions), type applications and abstractions, value variables (the language is left-to-right and call by value), and nat-literals. The formalization should be as described in class.

2. Implement the following in diMPL:
a) An ADT for queues. The ADT must allow queue elements to be anything (e.g., ints, bool→bool functions, etc), as long as all the queue elements are the same type.
b) A package that implements queues and exports the interface defined in Part (a).
c) A client that imports and uses—in some nontrivial but easy-to-understand way—the queue implementation defined in Part (b).

Finally,
d) Prove that your solutions to Parts (b) and (c) are well typed, and show how your solution to Part (c) evaluates at run time.