

## CS 515 Fall 2011 Homework 4 – Sample Solution

### 1.1)

The `let` and `let*` constructs can be simulated with lambda expressions. Consider the following procedure, which uses a `let` to define the named value `newval`:

```
(lambda (n)
  (let ((newval (+ n 2))
        (+ newval newval)))) ;; computes 2*(n+2)
```

To express this using only lambdas, make the named value the argument to an inner lambda abstraction whose body is the original function body (`+ newval newval`). Then apply this function to the body of the `let`, so that `(+ n 2)` is substituted everywhere for `newval`:

```
(lambda (n)
  ((lambda (newval) (+ newval newval)) (+ n 2)))
```

In a call-by-value interpreter, the body of the `let` would still be evaluated only once.

We can make `let*` by nesting the lambdas. So

```
(lambda (n)
  (let* ((newval (+ n 2))
         (newerval (+ newval 1)))
    (+ newerval newerval)))
```

would become

```
(lambda (n)
  ((lambda (newval)
    ((lambda (newerval)
      (+ newerval newerval)) (+ newval 1))) (+ n 2)))
```

Thus `let` can be seen as “syntactic sugar” for this type of expression, providing another demonstration of how everything can be done with lambdas.

### 1.2) See source file `VS-let.scm`