Keng Low
January 20, 2014

Induction

```
(define multiply
  (lambda ((a <number>) (b <integer>))
    (cond ((zero? b) 0)
          ((odd? b)
           (+ a (multiply (+ a a) (quotient b 2))))
          (else
           (multiply (+ a a) (quotient b 2))))))
```

Construct a proof using induction that shows this procedure works correctly. In other words, your proof should show that, for any number a and any integer b >= 0, that (multiply a b) returns the value ab.

```
For all a, and for all integers b >= 0,
(multiply a b) ==> {a * b}.
Proof of induction on b.

Assume (multiply a k) ==> {a * k} for all k < b
    (multiply a (k-1)) ==> {a * (k-1)}
                       ==> {ak - a}

If (k - 1) is odd and (k - 1) > 2:
    (+ a (multiply (+ a a) (quotient (k - 1) 2)))
    (+ a (multiply {2a} {k/2 - 1}))
    (+ a ak -2a)
    {ak - a}

If (k - 1) is even and (k - 1) > 2:
    (multiply (+ a a) (quotient (k - 1) 2))
    (multiply {2a} {k/2 - 1/2})
    (+ ak -a)
    {ak - a}

Base case

The inductive case handles all k, but when b = 0
(multiply k 0) ==>
(cond ((zero? b) 0 ...) ==>
    (zero? 0) ==> #t, so evaluate CONSEQUENT
==> {0}
```