Phillip booth
Question 31

(length (reverse xs)) = (length xs)

Base Case: xs is nil

={substitute simple-reverse}

(length (if null? ‘() ‘() (append (simple-reverse(cdr ‘()) (list1(car ‘())))))))

={null-empty law}

(length (if null? #t ‘() (append (simple-reverse(cdr ‘()) (list1(car ‘())))))))

={if-#t law}

(length ‘())

(length xs) when xs is null

Inductive Step: xs is guaranteed not to be null

(length (simple-reverse xs)) where xs =(cons z zs)

={using inductive step to show that xs does not equal nil}

(length (simple-reverse (cons z zs)))

={substitute in simple reverse}

(length (if null? (cons z zs) ‘() (append (simple-reverse zs) (list1 z))))))

={if -#f law}

(length (append (simple-reverse zs) (list1 z))))))

={length-append law}

(+ 1 (length(simple-reverse zs)) (length(list1 z))

={length –list1}

(+ 1 (length(simple-reverse zs)))

={indiction hypothesis}

(+ 1 (length zs))

={cons-length law}

(length (cons z zs))

(length xs)