Assign 4 Q 4

<u>Hypothesis</u>: reduce and reduce_tr will return the same result given the same arguments, for any chosen set of arguments.

<u>AXIOMS</u>

Axiom A:
op(base, m) = m

Axiom B:
op(n,m) = op(m,n)

Axiom C:
op(n, op(m,k))) = op(op(n,m), k)

<u>LEMMA</u>

Lemma: op(h, reduce(l, n, op)) = reduce_tr(l, op(h, n), op)

Proof by structural induction on l:

Base Case:

for any h, op, and n, let l = []

op(h, reduce([], n, op)) = reduce_tr([], op(h, n), op)

|-> op(h, n) = op(h, n)

Induction Hypothesis:

Op (h, reduce (l, n, op)) = reduce_tr (l, op (h, n), op)

Proof:

Show that op (h, reduce (a::l, n op)) = reduce_tr (a::l, op (h, n), op)

op (h, reduce (a::l, n op)) = reduce_tr (a::l, op (h, n), op) by definition of reduce and reduce_tr:

|-> op (h, op (a, reduce (l, n, op))) = reduce_tr (l, op (a, op (h, n)), op)

By axioms B and C:

|-> op (op (h, a), reduce (l, n, op)) = reduce_tr (l, op (op (h, a), n), op)

By induction hypothesis

|-> op (op (h, a), reduce (l, n, op)) = op (op (h, a), reduce (l, n, op))

<u>PROOF</u>

Proof by structural induction on a:

Base Case:

For any base and op, let a = []

reduce ([], base, op) = reduce_tr ([], base, op)

(fun reduce ([], base, op) = base) = (fun reduce_tr ([], base, op) = base)

Base = base

Induction Hypothesis:

Reduce (a, base, op) = reduce_tr (a, base, op)

Proof:

Show that reduce (b::a, base, op) = reduce_tr (b::a, base op)

By definition of reduce and reduce_tr

|-> op(b, reduce(a, base, op)) = reduce_tr (a, op(b, base), op)

By Lemma

|-> op(b, reduce(a, base, op)) = op(b, reduce(a, base, op))

This concludes the proof.