Boone Tison

Problem Set #3

1. [int, ()] collatzSeq(int n) {

int next;

if (n % 2 == 0) next = n/2;

else next = (3 \* n) + 1;

return list(next,

function() {

return collatzSeq(next);

}

);

}

2.

a. The procedure method cannot touch any data that the caller method is also using or touching.

b. If the caller method never uses the return value, then the caller can keep going while the function is also running. However, if the caller never uses the return data, then what is the pint of the return in the function, or even there could of been no purpose to calling the function. In addition, the function must not touch any data that the caller method is also using or touching.

3.

a. bool equivalent(Struct S1, S2) {

if S1 and S2 do not have the same number of fields, return false

for each field in S1 and S2 {

if f1 is a scalar

if f2 is not a scalar, return false

if f1 and f2 are not the same type, return false

if f1 is an array

if f2 is not an array, return false

if f1 and f2 are not the same size or have

different composition types, return false

if f1 is a pointer

if f2 is not a pointer, return false

if f1 and f2 types they point to are not equivalent,

return false

if f1 is a struct

if f2 is not a struct, return false

if equivalent(f1, f2) is false, return false

}

return true # Every field has a structurally equivalent type

}

b. bool equivalent(Struct S1, S2) {

if S1 and S2 do not have the same number of fields, return false

Put all the fields of S2 into a dict, where the key is the field name and the value is false

for each field in S1 {

for each key in the dict {

if the key's value is false and the key and field have the same type

key's value = true

}

if no key is found, return false

}

return true # Every field has been matched with a key

}

S1 and S2 are structurally equivalent. They have the same number of fields. They both have the structs S1 and S2. Then, the structs S3 and S4 are structurally equivalent.