CS 511 – Quiz 6: Sequential Erlang

17 October 2018

Names: Anthony Rusignuolo, Stephen Szemis

Pledge: I pledge my honor that I have abided by the Stevens Honor System.

Exercise 1

1. Implement a simple function `isComplete` that determines whether a binary tree is complete. Binary trees are represented as follows:

```
<btree> ::= {empty}
          | {node,<integer>,<btree>,<btree>}
```

It may be useful to use a queue. The queue operations are:

- `new() -> queue()`. Returns an empty queue.
- `is_empty(Q :: queue()) -> boolean()`. Tests if Q is empty and returns true if so, otherwise false.
- `in(Item, Q1 :: queue(Item)) -> Q2 :: queue(Item)`. Inserts `Item` at the rear of queue Q1. Returns the resulting queue Q2.
- `out(Q1 :: queue(Item)) -> {{value, Item}, Q2 :: queue(Item)} | {empty, Q1 :: queue(Item)}`. Removes the item at the front of queue Q1. Returns tuple value, Item, Q2, where Item is the item removed and Q2 is the resulting queue. If Q1 is empty, tuple empty, Q1 is returned.

For example,

```
1> Q0 = queue:new().
{[],[]}
2> queue:out(Q0).
{empty,{[],[]}}
3> Q1 = queue:in(2,queue:in(1,Q0)).
{[2],[1]}
4> queue:out(Q1).
{{value,1},{[],[2]}}
```

```
emptyCheck(Q) ->
  if is_empty(Q) ->
    true
  end,
  (i, Q1) = queue:out(Q),
  if (i =:= empty) ->
    false
  end,
  emptyCheck(Q1).


isCompleteHelper(Q) ->
  (i, Q1) = queue:out(Q)
  if (i == empty) ->
    emptyCheck(Q1)
  end,
  (val, left, right) = i,
  isCompleteHelper(queue:in(right, queue:in(left, Q1))).

isComplete(T) ->
  Q0 = queue:new()
```