Programming Paradigms Exam – 21 January 2020

Work Time: 2 hours

Default 1p

Problem 1 (3p): Please give a functional definition of {MemberTwice Xs Y} that tests whether Y is an element that occurs at least 2 times in Xs. For example, the call {MemberTwice [a b c d b e] b} should return true, whereas {MemberTwice [a b c] b} should return false.

Problem 2 (3p): Suppose that you are given a boolean expression described by a tree constructed from tuples as follows:
1. A boolean is described by a tuple boolval(N), where N is either true or false.
2. A logical *and* conjunction is described by a tuple booland(X Y), where both X and Y are boolean expressions.
3. A logical *or* disjunction is described by a tuple boolor(X Y), where both X and Y are boolean expressions.
Implement a function Eval that takes a boolean expression and returns its value. For example, booland(boolval(true) boolor(boolval(true) boolval(false))) is a boolean expression and its evaluation returns true.

Problem 3 (3p): The following is a naive attempt to write a concurrent Filter function:

```
fun {Filter L F}
 case L of
        X|Xs then if thread {F X} end
                        then X|{Filter Xs F}
                        else {Filter Xs F} end
       else nil
  end
end
```

Suggest how you may provide an alternative Filter operation with better concurrency. Outline the key steps that you need to make. (Hint : You may make use of non-declarative message-passing concurrency scheme.)