Lab Session Software Testing 2013, Week 4 With each deliverable, indicate the time spent.

1. If the course topic of this week was difficult for you, you should spend some time to read up on relations in Chapter 5 of "The Haskell Road".

   (Deliverable: List with specific points that cause difficulty, with a precise indication of what you do not understand, and why.)

2. Implement a random data generator for the datatype `Set Int`, where `Set` is as defined in `http://homepages.cwi.nl/~jve/rcrh/SetOrd.hs`.

   (Deliverables: Haskell program, indication of time spent.)

3. Implement operations for set intersection, set union and set difference, for the datatype `Set` defined in `http://homepages.cwi.nl/~jve/rcrh/SetOrd.hs`. Next, use automated random testing to check that your implementation is correct.

   Note: you may have to change `import List` to `import Data.List` in the module `SetOrd`.

   (Deliverables: Haskell program, test code, short test report, indication of time spent.)

4. Suppose we implement binary relations as list of pairs, Haskell type `[(a,a)]`.

   Assume the following definitions:

```
type Rel a = [(a,a)]

infixr 5 @@

(@@) :: Eq a => Rel a -> Rel a -> Rel a
r @@ s =
  nub [ (x,z) | (x,y) <- r, (w,z) <- s, y == w ]
```

   Use this to implement a function

```
trClos :: Ord a => Rel a -> Rel a
```

that gives the transitive closure of a relation, where the relation is represented as a list of pairs.

E.g., `trClos [(1,2),(2,3),(3,4)]` should give

`[(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)]`.

(Deliverable: Haskell program, indication of time spent.)

5. Test the function `trClos` from the previous exercise. Devise your own test method for this. Try to use random test generation. Define reasonable properties to test.

   (Deliverables: test code, short test report, indication of time spent.)