Test Report Assignment week 4

Assignment 4

For assignment 4, we built quickCheck-testable tests:

```
testSimpleIntersect :: Set Int -> Set Int -> Bool
testSimpleIntersect a b = subSet (setIntersect a b) a
  && subSet (setIntersect a b) b
```

This function tests if the intersect of a and b is a subset of a and b.

```
testSubsetUnion :: Set Int -> Set Int -> Bool
testSubsetUnion a b = subSet a (setUnion a b)
  && subSet b (setUnion a b)
```

This function tests if the a is a subset of the union of a and b, and if b is a subset of the union of a and b.

```
testDifferenceSubset :: Set Int -> Set Int -> Bool
testDifferenceSubset a b = subSet (setDiff a b) a
```

This function tests if the difference set of a and b is a subset of a.

```
testEquivalenceRelations :: Set Int -> Set Int -> Bool
testEquivalenceRelations a b = (setIntersect a b)
  == (setDiff (setUnion a b) (setUnion (setDiff a b) (setDiff b a)))
```

This function tests all three functions, by testing if the intersection of a and b is equivalent to the difference set of the union of a and b and the union of the difference set a b and b a. From these tests, we built an hspec function which tests the three properties for different sets of a and b. This quickCheck worked correctly.

Assignment 7

To test trClos we found on Wikipedia that it should hold, that:

$$R^+ \iff R \subseteq R^+ \cap R^+ is\, transative \cap R^+ is\, minimal$$

In the workshop we also proved that,

$$R \circ R \subseteq R \Leftrightarrow R\, is\, transative$$

We tested this using the following three methods:

```
testTrClosRoRinR :: [(Int, Int)] -> Bool
testTrClosRoRinR r = testTransitive $ trClos r
```

Which tests $R \circ R \subseteq R \Leftrightarrow R$ is transitive. Here, testTransitive is defined as:

```
testTransitive :: [(Int, Int)] -> Bool
testTransitive r = (r @@ r) `allIn` r
```

And `allIn` is defined as following:

```
allIn :: Eq a => Rel a -> Rel a -> Bool
allIn [] s = True
allIn (r:rs) s = r `elem` s && rs `allIn` s
```

We then tested if R was in the transitive closure of R, by calling the following function:

```
testTrClosRinR :: [(Int, Int)] -> Bool
testTrClosRinR r = r `allIn` (trClos r)
```

And finally we tested that R^+should be minimal, so there shouldn't be a smaller subset thats also transative and contains R, we solved this with an $O(2^n)$ time test function, which basically creates al the possible powerLists of R^+ and test if there is only one possibility which is transitive.

```
testTrClosIsMinimal :: [(Int, Int)] -> Bool
testTrClosIsMinimal rNotUnique = (1==) $ length $
  (filter testTransitive (map (r ++) (powerList $ ((trClos r) \\ r))))
  where r = nub rNotUnique
```

From these tests, we built an hspec function which tests the three properties for different sets of a and b. This quickCheck worked correctly. This is shown in assignment 6.