

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 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 $a \setminus b$ and $b \setminus a$. 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:

$$R^+ \iff R \subseteq R^+ \cap R^+ \text{ is transitive} \cap R^+ \text{ is minimal}$$

In the workshop we also proved that,

$$R \circ R \subseteq R \iff R \text{ is transitive}$$

We tested this using the following three methods:

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

Which tests $R \circ R \subseteq R \iff R$ is transitive:

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

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

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

Here, '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
```

And than R^+ should be minimal, so there shouldn't be a smaller subset that's also transitive and contains R , we solved this with an $O(2^n)$ time test function, which basically creates all the possible powerLists of R^+ .

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

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.