

Lab session Software Testing, week 4

Fabien Tesselaar, Lulu Zhang, Tim Gosen
September 2013

Exercise 5:

In this report, we use random test generation to test our `trClos` function.

Time spent: 1 hour

First, we use a function to generate random relations, such as a relation named “`r`”. Then, we compare the result of “`r @@ r`” with “`trClos r`”. If `trClos` function is right, the result of this should be a transitive closure, which means `r @@ r` should be a part of that result. If not, this function is false.

The code is:

```
getRandomRel :: IO (Rel Int)
```

```
getRandomRel = do
```

```
    d <- getRandomInt 10
```

```
    t <- getRandomInt 5
```

```
    getRandomR d t
```

```
getRandomR :: Int -> Int -> IO (Rel Int)
```

```
getRandomR _ 0 = return []
```

```
getRandomR d t = do
```

```
    m <- getRandomInt d
```

```
    n <- getRandomInt d
```

```
    fs <- getRandomR d (t-1)
```

```
    return ((m, n):fs)
```

```
testTrClos :: IO()
```

```
testTrClos = do
```

```
    r <- getRandomRel
```

```
    if subSet (list2set (r @@ r)) (list2set (trClos r))
```

```
        then print (show r ++ "--the transitive closure: " ++ show (trClos r) ++ "--@@: "  
++ show(r @@ r) ++ " --True" )
```

```
        else error (show r ++ "--the transitive closure: " ++ show (trClos r) ++ "--@@: " ++ show(r  
@@ r) ++ " --False")
```

Here are some test results:

Input	Output1 (trClos)	Output2 (@@)	Result
[(1,6),(0,2),(5,7)]	[(1,6),(0,2),(5,7)]	[]	TRUE
[(7,1),(4,6),(9,4),(3,2)]	[(7,1),(4,6),(9,4),(3,2)]	[(9,6)]	FALSE
[]	[]	[]	TRUE
[(1,2),(2,1)]	[(1,2),(1,1),(2,1)]	[(1,1),(2,2)]	FALSE
[(1,1),(0,1)]	[(1,1),(0,1)]	[(1,1),(0,1)]	TRUE
[(3,3),(0,1),(2,0),(2,2),(0,3)]	[(3,3),(0,1),(2,0),(2,3),(2,2),(0,3)]	[(3,3),(2,1),(2,3),(2,0),(2,2),(0,3)]	FALSE
[(2,5)]	[(2,5)]	[]	TRUE
[(4,5),(0,1),(5,1)]	[(4,5),(4,1),(0,1),(5,1)]	[(4,1)]	TRUE