Hvordan skrive en feilende test om til jUnit test

Bjarte Johansen

May 14, 2018

La oss si at vi holder på å kjøre testen `add_entry_returns_previous_value`

```
1  /**
2   *  Check that add(Entry) returns the previous value
3   */
4  public Property add_entry_returns_previous_value() {
5      return property(isKVList, arbString,
6              (kvs, value) -> implies(kvs.length() > 0 && !kvs.exists(kv -> value.equals(kv._2())),
7                      // Choose a random index in the key-value list.
8                      () -> property(choose(0, kvs.length()-1),  i -> {
9                          P2<Integer, String> entry = kvs.index(i);
10                         SortedTreeMap<Integer, String> tm = new SortedTreeMap<>(intOrd.toComparator());
11                         kvs.foreachDoEffect(kv -> tm.add(kv._1(), kv._2()));
12                         String old_value = tm.add(new Entry<>(entry._1(), value));
13                         return prop(old_value.equal(entry._2()));
14                     })
15             )
16     );
17 }
```

og får den følgende feilen

```
java.lang.Error: Exception on property evaluation with arguments:
   List(List((-1,neg)),mzyw,0)
java.lang.NullPointerException
at fj.Equal$2.equal(Equal.java:196)
at fj.Equal.eq(Equal.java:81)
at main.java.no.uib.info233v18.oblig4.SortedTreeMapTest.lambda$null$16(SortedTreeMapT
at fj.P1.lambda$null$2(P1.java:81)
at fj.P1$$Lambda$20.f(Unknown Source)
at fj.P$2._1(P.java:76)
at fj.P1.f(P1.java:23)
        ....
```

Det feilen sier er at listen over argumentene som ble prøvd, som førte til feilen er `List(List((-1,neg)),mzyw,0)`. Den har en liste av par som

skal legges til, og en ny verdi for elementet i listen av par. Altså, den testen prøvde å legge til nøkkelen -1 med verdien "neg", for å å erstatte verdien til det første elementet i listen av par (-1) med "mzyw". Dette førte til en "NullPointerException.

Hvis jeg ønsker å skrive dette om til en jUnit test for å kunne gå stegvis igjennom den feilen så kunne jeg skrevet

```
1  @Test
2  void one_entry_replace() {
3    SortedTreeMap<Integer, String> tm = new SortedTreeMap<>(Comparator.naturalOrder());
4    tm.add(-1, "neg");
5
6    String old_value = tm.add(new Entry<>(-1, "mzyw"));
7    assertNotNull(old_value);
8    assertEquals("neg", old_value);
9  }
```

Jeg kan da bruke debuggeren i intellij til å stegvis gå igjennom beviset jeg har fått fra quickcheck for å se hvorfor `tm.add(Entry<K,V>)` returnerer `null`.