Test Report Week 2, Assignm. 1 Software Testing

Implementation

For each different outcome for the *triangle* function, we developed two tests:

1. If the conditions for the statement of the type under test holds, it should return the type under test.
2. If the conditions for the statement of the type under test do not hold, it should not return the type under test.

The tests are based on the following statements:

1. The shape triangle($x \in \mathbb{N}, y \in \mathbb{N}, z \in \mathbb{N}$) is not a triangle (NoTriangle) in case x, y or z is lower than or equal to 0, thus:

   $$triangle(x, y, z) = \text{NoTriangle} \rightarrow (x \leq 0 \vee y \leq 0 \vee z \leq 0)$$

   $$triangle(x, y, z) \neq \text{NoTriangle} \rightarrow \neg(x < 0 \vee y \leq 0 \vee z \leq 0)$$

2. The shape triangle($x \in \mathbb{N}, y \in \mathbb{N}, z \in \mathbb{N}$) is Equilateral in case all edges are of the same length, thus:

   $$x, y, z \in \mathbb{N} \wedge triangle(x, y, z) = \text{Equilateral} \rightarrow (x = y = z)$$

   $$x, y, z \in \mathbb{N} \wedge triangle(x, y, z) \neq \text{Equilateral} \rightarrow \neg(x = y = z)$$

3. The shape triangle($x \in \mathbb{N}, y \in \mathbb{N}, z \in \mathbb{N}$) is Rectangular, in case one side is $\frac{1}{2}\pi$ rad, thus, by Pythagoras:

   $$x, y, z \in \mathbb{N} \wedge triangle(x, y, z) = \text{Rectangular} \rightarrow (x^2+y^2 = z^2 \vee x^2+z^2 = y^2 \vee z^2+y^2 = x^2)$$

   $$x, y, z \in \mathbb{N} \wedge triangle(x, y, z) \neq \text{Rectangular} \rightarrow \neg(x^2+y^2 = z^2 \vee x^2+z^2 = y^2 \vee z^2+y^2 = x^2)$$

4. The shape triangle($x \in \mathbb{N}, y \in \mathbb{N}, z \in \mathbb{N}$) is Isosceles, in case two edges are of the same size and not all edges are, thus:

   $$x, y, z \in \mathbb{N} \wedge triangle(x, y, z) = \text{Isosceles} \rightarrow (\neg(x = y = z) \wedge (x = y \vee x = z \vee y = z))$$

   $$x, y, z \in \mathbb{N} \wedge triangle(x, y, z) \neq \text{Isosceles} \rightarrow \neg(\neg(x = y = z) \wedge (x = y \vee x = z \vee y = z))$$

5. The fifth test is basically a check if the above statements are incorrect, without the regard of Equilateral, as it is basically an extension of Isosceles (i.e. weaker constraints).

All these tests were implemented as subsets of the mathematical statements (due to infinite lists) using list comprehensions in Haskell, returning a set of booleans which should all be true in case of success, which then concludes the function (a boolean) through a call of VALIDATEALL, which checks if all booleans in the specified argument are true:

```
validateAll :: [Bool] -> Bool
validateAll [] = True
validateAll (x:xs)
        | x = validateAll xs
        | otherwise = False
```

Then, there is a function that runs all tests and checks if all the tests returned true: TESTTRIANGLE.

```
testTriangle :: Bool
testTriangle = validateAll
[ testTriangle1a, testTriangle1b,
  testTriangle2a, testTriangle2b,
  testTriangle3a, testTriangle3b,
  testTriangle4a, testTriangle4b,
  testTriangle5a, testTriangle5b ]
```

Results

The results of all functions (testTriangle (and all seperate tests as well)), the results were all True.