

Test report assignment 2

We tested if all generated problems were minimal by creating a function *isMinimal*, which tries to remove one *filled* number at a time and checks if it is still a unique function. If, for all cases, this is false, but for the case where no number is removed, then we can conclude that the solution is *minimal*, because for each removed number, the solution is not unique anymore.

For the testing of the specifications using QuickCheck in the first assignment, we already defined the test for this function:

```
prop_min = monadicIO $ do [r] <- run $ rsolveNs [emptyN]
                        p <- run $ genProblem r
                        assert $ isMinimal p
```

which asserts that for each arbitrary problem, the *isMinimal*-property holds.

The isMinimal function is defined as following:

```
isMinimal n = uniqueSol n && all (isNotUniqueWithout n) (filledPositions $ fst n)
```

Thus it checks that the original NODE n is unique, and for all filled positions, when removed, is not unique anymore, using the function ISNOTUNIQUEWITHOUT N X, which is defined as following:

```
isNotUniqueWithout n x = (not $ uniqueSol $ eraseN n x )
```

The function ERASEN removes the value at location x and then tests if the uniqueness constraint still holds.

We tested *prop_min* using QuickCheck as following:

```
*Lab5> quickCheck prop_min
```

Which returned:

```
+++ OK, passed 100 tests.
```

Thus, we can conclude that all generated problems (at least 100 of them) are in fact minimal.