

Assignment Project Exam Help

COMP9141

Software System Design and Implementation

<https://powcoder.com>

Property Based Testing Practice

Curtis Millar

CSE, UNSW (and Data61)

17 June 2020

Add WeChat powcoder

Exercise 1

Assignment Project Exam Help

- 1 **Simple Picture:** add the chimney and smoke
- 2 **Moving Objects:** implement `movePictureObject`
- 3 **Generating a Picture:** generate pictures of circles using `simpleCirclePic`

<https://powcoder.com>
Add WeChat powcoder

Property Based Testing

Assignment Project Exam Help

Key idea: Generate random input values, and test properties by running them

Example (QuickCheck Property)

```
prop_reverse pp xs ys =  
  reverse (xs ++ ys) == reverse ys ++ reverse xs
```

Haskell's *QuickCheck* is the first library ever invented for property-based testing. The concept has since been ported to Erlang, Scheme, Common Lisp, Perl, Python, Ruby, Java, Scala, F#, OCaml, Standard ML, C and C++.

<https://powcoder.com>

Add WeChat powcoder

Mersenne Prime Example

Example (Demo Task)

- The n^{th} Mersenne number $M_n = 2^n - 1$.
- M_2 , M_3 , M_5 and M_7 are all prime numbers.
- **Conjecture:** $\forall n, \text{prime}(n) \Rightarrow \text{prime}(2^n - 1)$

<https://powcoder.com>

Let's try using QuickCheck to answer this question.

After a small number of guesses and fractions of a second, QuickCheck found a counter-example to this conjecture: 11.

It took humanity about two thousand years to do the same.

Add WeChat powcoder

Semigroup and Monoid Properties

Last week we proved by hand that `[]` forms a semigroup with `++` as its associative operator and a monoid with `[]` as its identity element.

We can show the same properties much faster (although less completely) with property based testing.

QuickCheck Properties

-- Semigroup laws

```
prop_listAssociative xs yz zs = ((xs ++ ys) ++ zs) == (xs ++ (ys ++ zs))
```

-- Monoid laws

```
prop_listLeftIdentity xs = xs == [] ++ xs
```

```
prop_listRightIdentity xs = xs == xs ++ []
```

<https://powcoder.com>

Add WeChat powcoder

Reverse Involution

Assignment Project Exam Help

Last week we also proved by hand that the reverse function is an *involution*.

This took over twenty minutes.

Let's see how long it takes QuickCheck

<https://powcoder.com>

QuickCheck Property

```
prop_reverseInvolution xs = reverse (reverse xs) == xs
```

Add WeChat powcoder

Ransom Note Example

Assignment Project Exam Help

Example (Demo Task)

Given a magazine (in String form), is it possible to create a ransom message (in String form) from characters in the magazine.

`canMakeRansom :: RansomNote -> Magazine -> Bool`

- ❶ Write a specification
- ❷ Create an efficient implementation
- ❸ Test the implementation

In Haskell.

Add WeChat powcoder

<https://powcoder.com>

Graphics

Assignment Project Exam Help

Write some specifications for the following functions, use them to create properties, and then test an implementation.

- 1 Horizontal flip
- 2 Vertical flip
- 3 Rotate 180 degrees

<https://powcoder.com>

Example (Demo Task)

Implement the above for a single `Path`. (You might want to try and implement these for other `PictureObject` constructors or for an entire `Image` as self-practice.) In Haskell.

Add WeChat powcoder

Proofs

Assignment Project Exam Help

Proofs:

- Proofs must make some assumptions about the environment and the semantics of the software.
- Proof complexity grows with implementation complexity, sometimes drastically.
- If software is **incorrect**, a proof attempt might simply become stuck: we do not always get constructive negative feedback.
- Proofs can be labour and time intensive (\$\$\$), or require highly specialised knowledge (\$\$\$).

<https://powcoder.com>

Add WeChat powcoder

Testing

Assignment Project Exam Help

Compared to proofs

- Tests typically run the actual program, so requires fewer assumptions about the language semantics or operating environment.
- Test complexity does not grow with implementation complexity, so long as the specification is unchanged.
- Incorrect software when tested leads to immediate, debuggable counterexamples.
- Testing is typically cheaper and faster than proving.
- Tests care about **efficiency** and **computability**, unlike proofs.

We **lose** some assurance, but **gain** some convenience (\$\$\$).

Verification versus Validation

Assignment Project Exam Help

"Testing shows the presence, but not the absence of bugs."
– Dijkstra (1969)

Testing is essential but is insufficient for safety-critical applications.

Add WeChat powcoder

Homework

Assignment Project Exam Help

- ① Last week's quiz is due on Friday. Make sure you submit your answers.
- ② The second programming exercise is due by the start of my next lecture (in 7 days).
- ③ This week's quiz is also up, it's due next Friday (in 9 days).

<https://powcoder.com>
Add WeChat powcoder

Consultations

Assignment Project Exam Help

Tomorrow, 9am to 11am on Blackboard Collaborate
<https://powcoder.com>
link on course website.

Be ready to share your screen with REPL (`ghci` or `stack repl`) and editor set up.

Add WeChat powcoder