Lab 2: Invariants

COSC 3020: Algorithms and Data Structures

Lars Kotthoff
larsko@uwyo.edu

18 September 2019

Instructions

Attempt to finish the tasks below during the lab time. You have until Friday, 20 September 2019, 23:59h to submit the solutions to WyoCourses. You may ask your TA for feedback before submitting, but this feedback will be qualitative only.

You may *not* use external libraries in your code unless explicitly stated.

1 Fibonacci

Recall the definition of the Fibonacci series: the first two numbers are 1, each subsequent number is the sum of the two numbers preceding it. Implement a function that computes the Fibonacci numbers recursively, storing the results in an array.

For example, the return value of `fib(7)` is the following array:

index	0	1	2	3	4	5	6
value	1	1	2	3	5	8	13

Submit your complete code, including a function that demonstrates that your implementation works as expected with a few test inputs.

Total 6 points.

2 Invariant

What is a good invariant for the recursive implementation of `fib()`, i.e. something that is always true at the beginning of the recursive call?

Hint: Think about what the "state of the world" is here and what you can say about it at the start of each recursive call.

Describe your reasoning and the conclusion you've come to. Your reasoning is the most important part.

Total 4 points.

Testing

You can use `jsverify` (`https://jsverify.github.io/`) to automatically test your code. Assuming that your code is in a file `code.js`, you can use the following code to test it:

```
const fs = require('fs');
const jsc = require('jsverify');

eval(fs.readFileSync('code.js')+'');

function fibTest(n) {
    var fib_solns = [];
    if(n == 0) return fib_solns;
    fib_solns[0] = 1;
    if(n == 1) return fib_solns;
    fib_solns[1] = 1;
    if(n == 2) return fib_solns;
    for(var i = 2; i < n; i++) fib_solns[i] =
        fib_solns[i-1] + fib_solns[i-2];
    return fib_solns;
}

const test =
    jsc.forall("nat", function(n) {
        return JSON.stringify(fib(n)) ==
            JSON.stringify(fibTest(n));
    });
jsc.check(test);
```

This is optional, but highly encouraged. How exactly to get this code to run is left as an exercise to the reader. No points, but the satisfaction of knowing how to do property-based testing.