# 15-819M Data, Code, Decisions

**Assignment 4 ($\sum 50$) due by Thu 11/19/2009**

André Platzer

Carnegie Mellon University, Computer Science Department, Pittsburgh, PA aplatzer@cs.cmu.edu

Disclaimer: No solution will be accepted that comes without an **explanation**!

## Exercise 1 JML-KeY Verification (25p)

Complete the JML specification in the following class. Then verify in KeY that the gcd function in fact returns the greatest common divisor.
Hint: x % y is the modulo function in KeY.
Proof hint: Use the settings DefOps for Arithmetic treatment. It is probably a good idea to first prove gcd(int,int) using a contract for method gcdHelper, and then go on proving the required properties of gcdHelper in Proof → Show Used Specifications.

```
public class Euclid {
    /*@
      @ public normal_behavior  ....;
      @*/
    public static int gcd(int a, int b) {
        if (a < 0) a = -a;
        if (b < 0) b = -b;
        int big, small;
        if (a > b) {
            big = a;
            small = b;
        } else {
            big = b;
            small = a;
        }
        return gcdHelper(big, small);
    }

    /*@
      @ public normal_behavior ...;
      @ assignable \nothing;
      @*/
    private static int gcdHelper(int _big, int _small) {
        int big = _big;
        int small = _small;
        /*@
          @ loop_invariant ...;
          @ decreases ...;
          @ assignable ...;
          @*/
        while (small != 0) {
            final int t = big % small;
            big = small;
            small = t;
        }
        return big;
    }
}
```

## Exercise 2 Freestyle KeY Verification (25p)

Choose an interesting algorithm and at least one interesting property of it. Give formal specifications of the algorithm, either in Java Dynamic Logic of KeY or in JML. Verify the property in KeY.
Evaluation: The points scored for this exercise depend on how smart and interesting your choice of algorithm and proof is. Fully verified but fully trivial algorithms score less than more interesting algorithms for which only some of the relevant properties have been proven. Extremely interesting algorithms without proofs score 0.