Program Verification Prof. Dr. H. Peter Gumm
Summer 2023

Exercise sheet 04

Deadline: May 20, 8:00 p.m.

This time, you should submit 2 files. The first file, **ex04**_*your_name*.**dfy**, should be a Dafny-file containing the answers to Problems 1 and 2. The second file should be a pdf-file **ex04**_*your_name*.**pdf**, containing your solution to problems 3 and 4.

If you do not know how to create pdf-files containing mathematical/logical formulas, please alert me Monday morning. I will take some time off the lecture and show you various ways how to do that.

Problem 1. In class we discussed a simple program for calculating the sum of the first n odd numbers. Below, I modified it slightly, mainly switching the assignments in the loop's body. At the places marked with P, I1,and I2, add appropriate conditions so that the program verifies in Dafny.

```
method sumOdds(n:nat) returns (sum:nat)
requires  /*  P  */
ensures sum == n*n;
{
  sum := 1;
  var  i := 0;
  while i < n-1
    invariant    /* I1 */
    invariant    /* I2 */
  {
    i := i+1;
    sum := sum + 2*i+1;
  }
}
```

(3 points)

Problem 2 (Division by repeated subtraction)**.** Write a method $intDiv(n, d)$ which takes positive integers n and d and yields integer quotient q and remainder r. For instance, if $n = 17$,and $d = 3$ then $q = 5$ and $r = 2$, since $17 = 5 * 3 + 2$. You are to specify and write a method $intDiv(n : int, d : int)$ which keeps subtracting d from n until the result falls below d. In this exercise, do **not** use „/“ or „%“.

(a). Write the specification in Dafny, i.e. equip the header $method\ intDiv(n : int, d : int)\ returns\ (q : int, r : int)$ with appropriate *requires* and *ensures* clauses.

(b). Implement *intDiv* as specified in the first step. Notice that it is not allowed in Dafny to modify the input parameters. In this case, it means that you may not modify n or d. It is, however, allowed to modify the output variables, in this case: q and r. In fact, you need not not any further variables in your program.

(c). Add invariants to your program so that Dafny accepts it as verified.

(1+1+3 points)

Problem 3 (Floyd follows from Hoare)**.** (a). Let R be an arbitrary logical proposition, v a variable, and t a term. Under which condition is $(\exists x_0.R)[t/v] = \exists x_0.R[t/v]$?

(b). Give an example where equality does **not** hold.

(c). Using only Hoare's rule $\frac{P \to Q[t/v]}{\{P\} v := t \{Q\}}$, derive Floyd's rule $\{P\} v := t \{\exists v_0.(P[v_0/v] \wedge v = t[v_0/v])\}$. Justify every step.

(1+1+3 points)

Problem 4 (if-then-else)**.** (a). Do exercise 2.24 from the book.

(b). Some languages possess ***if-then-elsif-else*** commands with syntax: **if** B_1 **then** C_1 **elsif** B_2 **then** C_2 **else** C_3. Formulate an appropriate Hoare-rule for such commands:

$$\frac{??}{\textbf{if } B_1 \textbf{ then } C_1 \textbf{ elsif } B_2 \textbf{ then } C_2 \textbf{ else } C_3}$$

(1+2 points)