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

Exercise sheet 06

Deadline: June 03, 8:00 p.m.

Please submit a Dafny-file **ex06_*your_name*.dfy** for Problem 1 and a pdf-file **ex06_*your_name*.pdf**, for problems 2 and 3.

Problem 1 (3+2 points)**.** $gcd(m,n)$ is the greatest common divisor of two natural numbers m and n. A possible definition as a Dafny function is:

```
ghost function gcd(x:int,y:int):int
  requires x > 0 && y > 0
{
  if x==y then x
  else if x > y then gcd(x-y,y)
  else gcd(x,y-x)
}
```

(a). Implement a method `gcdI(m:int,n:int) returns (d:int)` to compute $gcd(m,n)$ iteratively, using a while-loop. Your method should have appropriate `invariant, requires` and `ensures` clauses, including at least `ensures d == gcd(m,n)`. Your while loop will probably need a `decreases` clause which specifies an integer term that gets smaller with each pass through the loop.

(b). If you modify the above *gcd*-function by replacing `else gcd(x,y-x)` in the last line by `else gcd(y,x)`, then the new `gcd` will require a `decreases` clause. Which one will work? Check it in Dafny by renaming the new version of `gcd` to `gcd'`.

Problem 2 (4+4 pts)**.** We have learned Hoare's rule for a while loop:

$$\frac{P \to I,\ \{I \wedge B\}C\{I\},\ I \wedge \neg B \to Q}{\{P\}\textbf{while}\ B\ \textbf{do}\ C\ \{Q\}} \qquad \text{(Hoare's rule.)}$$

The following rule was suggested by Silas Brown:

$$\frac{P \wedge B \to R,\ \{R\}\,C\,\{(B \wedge R) \vee (\neg B \wedge Q)\},\ P \wedge \neg B \to Q,}{\{P\}\textbf{while}\ B\ \textbf{do}\ C\ \{Q\}} \qquad \text{(Brown's rule)}$$

The task is to show that both rules are equivalent.

(a). Brown's rule implies Hoare's rule.

  (a) Given P, B, Q, R, what could you choose as invariant I?
  (b) Prove Hoare's rule, assuming Brone's rule

(b). Show that from Hoare's rule one can derive Brown's rule

  (a) Given P, B, Q, I, what could you choose for R as required in Brown's rule?
  (b) Prove Brown's rule, assuming Hoare's rule.

Problem 3 (3 pts)**.** Calculate all verification conditions generated by the following annotated specification. Are they valid?

$\{x = n\}$
`y:=1;`
while `x!= 0` **do**
    invariant $x! * y = n!$
  `y:=y*x;`
  `x:=x-1`
$\{x = 0 \wedge y = n!\}$