Foundations of Computer Software: Exercise 5

February 3, 2012

Exercise 5-1

1. Save the following program as "div.mlw".

```
module Div
use import int.Int

let rec div (m:int) (n:int) variant {m} =
  { m >=0 /\ n > 0 }
  if m < n then
     0
  else
     1 + div (m-n) n
  {exists r:int. (0 <= r < n /\ m = result*n + r)}
end
```

2. Run the following command, and check that a Coq file `div_Div_WP_parameter_div.v` has been generated.

```
why3ml -P coq -o . div.mlw
```

3. Read `div_Div_WP_parameter_div.v` from Coqide, and complete the proofs of the verification conditions.

Exercise 5-2

1. Write an annotated program `mod` to compute the remainder of `m` divided by `n`, and generate verification conditions using `why3ml`.
2. What does each part of the verification conditions means, and which part of the program does it come from?
3. Prove the verification conditions by using Coq.

Exercise 5-3 (Advanced)

Write a program `gcd.mlw` to compute the greatest common divisor of two positive integers, and verify it by using why and Coq.

In the program, you also need to provide the definition of the greatest common divisor. So, gcd.mlw should look like:

```
module Gcd
use import int.Int

predicate isCD (m n c: int) =
  (** c is a common divisor of m and n **)
  (exists a b:int. m = a*c /\ n = b*c)
predicate isGCD (m n g:int) =
  (**  g is the greatest common divisor **)
  isCD m n g /\
  forall c:int. (isCD m n c -> c <= g)

let rec gcd (m:int) (n:int) variant {...} =
 (** actual code for gcd **)
  ...
```