Exercise Sheet 6: Specification and Verification with Higher-Order Logic (Summer Term 2010)

Date: 31.05.2010

Exercise 1 Inductive Definitions, Lattices and Fixpoints

a) (Prepare!) Define the reflexive, transitive closure of a relation $r$ as inductive set.

b) (Prepare!) Define a function whose least fixpoint is the aforementioned set.

c) (Prepare!) Let $L$ be a complete lattice, $a, b \in L$ and $a \leq b$. Prove that the closed interval $[a, b]$ is a complete lattice.

Reminder: $[a, b] := \{ x. a \leq x \leq b \}$

It is not required that you solve this exercise in Isabelle/HOL.

Exercise 2 Case Study: Greatest Common Divisor

a) Consider the following implementation of the greatest common divisor function:

fun gcd :: "nat => nat => nat" where
  "gcd m 0 = m" |
  "gcd m n = gcd n (m mod n)"

Prove that the function really computes the greatest common divisor of $m$ and $n$.

It might be useful to define and prove the following properties of $gcd$ first:

- The result of $gcd$ divides both arguments.
- Each common divisor divides the result of $gcd$.
- Each divisor of the result of $gcd$ is a common divisor.
- The result of $gcd$ is not zero if at least one argument is not zero.

Hint: In Isabelle/HOL, the property that $a$ divides $b$ is expressed by: $a \ dvd \ b$.

b) Prove the following property of $gcd$: $k \cdot gcd \ m \ n = gcd (k \cdot m) (k \cdot n)$.

c) Consider a slightly different implementation of the greatest common divisor function:

fun gcd :: "nat => nat => nat" where
  "gcd m n = (if n = 0 then m else gcd n (m mod n))"

- Prove that this implementation is equivalent to the first one.
- Prove the property of b) for this implementation.

d) Use the main property of a) to define the greatest common divisor non-recursively with the Hilbert-Choice operator (SOME), i.e. not using the Euclidean algorithm.

Prove the equivalence of this function to the original $gcd$. 