Prof. Dr. A. Poetzsch-Heffter
Dipl.-Inf. P. Michel
Dipl.-Inf. C. Feller

University of Kaiserslautern
Department of Computer Science
Software Technology Group

Exercise Sheet 7: Specification and Verification with Higher-Order Logic (Summer Term 2012)

Date: 13.06.2012

Exercise 1 Case Study: Quicksort

In the lecture you have seen an elegant way to specify and prove quicksort (see “`QuickSort.thy`”).

The file “`EQSort.thy`” contains another – more efficient – version of quicksort, which calculates the split with the pivot element in one pass through the list. The file also contains all specifications and proofs for its correctness.

a) Go through the *efficient* version of the specification, model and proof and compare them to the elegant one. In particular, note how the splitting is done and what that means for the proofs.

b) Download the theory “`RQSort.thy`” – which stands for *refined* quicksort – in which you prove the correctness of the *efficient* quicksort. To achieve this,

   1. prove that the efficient version of quicksort is equivalent to the elegant one and then
   2. prove the two main properties by using the theorems from the elegant version.

Exercise 2 Case Study: Mergesort

In this exercise we take a look at another sorting algorithm, namely *mergesort*. We are thereby changing the model, but keep the properties we already specified and validated for quicksort.

Download the “`MSort.thy`” theory from our website, which contains all the necessary definitions and properties.

Complete all the missing proofs and find the necessary helper lemmas, which capture the basic ideas of the algorithm.

Exercise 3 Inductive Definitions, Lattices and Fixpoints

This exercise is meant to deepen your understanding of inductive sets and the theory they are based on. You should do all of the following assignments *on paper* only.

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!) To get to know the definitions of lattice, complete lattice, supremum, infimum, etc. it is useful to do some proofs of simple properties involving them. For example, you should be able to do the following proofs:

   1. Let $P(S)$ be the power set of a set S. Proof that $(P(S), \subseteq)$ is a complete lattice.
   2. Proof that every closed interval (Def. I) of a complete lattice is a complete lattice.
   3. The definition of *complete lattice* from the slides is not actually based on *lattices*, but on partially ordered sets. Look at the definition of lattices based on partial orders (Def. L1) and convince yourself that every complete lattice is also a lattice.
   4. An equivalent definition of lattice from algebra (Def. L2) is not based on partial orders. Proof that you can define such a lattice $(S, \wedge, \vee)$ for every complete lattice $(S, \leq)$.
   5. Proof that for every lattice (Def. L2) both operations are idempotent ($x \wedge x = x$ and $x \vee x = x$).

   Definitions:

   (I) Let $(S, \leq)$ be a partial order. The *closed interval* for $a, b \in S$ is defined as: $[a, b] := \{x.\ a \leq x \leq b\}$

   (L1) A partially ordered set $(S, \leq)$ is a *lattice* iff

   $$\forall x, y \in S : \{x, y\} \text{ has both a supremum and an infimum in } S$$

   The normal notation for these two elements is $a \vee b$ and $a \wedge b$, respectively.

   (L2) A set $(S, \wedge, \vee)$ with two inner binary operations $\wedge$ and $\vee$ is a *lattice* iff both operations are associative, commutative and are connected by the absorption law ($x \wedge (x \vee y) = x$ and $x \vee (x \wedge y) = x$).