Prof. Dr. K. Madlener
Dipl.-Inf. P. Michel
Dipl.-Inf. C. Feller

University of Kaiserslautern
Department of Computer Science
AG Grundlagen der Informatik

Exercise Sheet 7: Specification and Verification with Higher-Order Logic (Summer Term 2011)

Date: 22.06.2011

Exercise 1 Case Study: Quicksort

In the lecture you have seen an elegant way to specify and prove quicksort. You can download the "`QSort.thy`" (and "`Universe.thy`") containing this version.

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. Describe why the proofs are more complicated in *efficient* version.

b) Create a new theory "`RQSort.thy`" – which stands for *refined* quicksort – in which you prove the correctness of the *efficient* quicksort by proving an equivalence to the *elegant* version. Use the specification of "`QSort.thy`" to formulate the two correctness theorems.

   Hint: Instead of importing Main in "`RQSort.thy`" *you should import both* "`QSort.thy`" *and* "`EQSort.thy`". *You then have to qualify each symbol you use from these theories to make clear which you are talking about.*

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 (together with "`Universe.thy`").

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