

Formal Specification of Abstract Datatypes

Exercise 4 (June 9)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

The result for each exercise is to be submitted by the deadline stated above via the Moodle interface as a .zip or .tgz file which contains

- a PDF file with
 - a cover page with the title of the course, your name, Matrikelnummer, and email-address,
 - the content required by the exercise (specification, source, proof),
- (if required) the CafeOBJ (.mod) file(s) of the specifications.

Exercise 4: Specification of Queues

A *queue*¹ is a “First In/First Out” data structure with operations *empty* (the queue without any elements), *isempty* (is the queue empty?), *enqueue* (add an element to the tail of the queue), *dequeue* (delete an element from the head of the queue), *head* (return the element at the head of the queue).

1. Write a loose specification with (possibly free) constructors of the abstract datatype *Queue* in a logic of your choice. You may assume that queue elements are values of sort *Nat*.
2. Similarly, write an initial specification of the abstract datatype *Queue* in conditional equational logic.
3. Compare the specifications and discuss (informally) their differences. Are the specifications strictly adequate with respect to the classical algebra of queues of natural numbers? Why do you think so (for answering these questions, assume that in the classical algebra $head(empty) = 0$ and $dequeue(empty) = empty$)?.
4. Implement the initial specification in CafeOBJ and test it with some reductions.

¹[http://en.wikipedia.org/wiki/Queue_\(data_structure\)](http://en.wikipedia.org/wiki/Queue_(data_structure))