

Formal Specification of Abstract Datatypes

Exercise 4 (June 11)

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 an unspecified sort **Elem**.
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 (assuming that **Elem** is strictly generated)? Why do you think so?
4. Implement the initial specification in CafeOBJ (using **Nat** for **Elem**) and test it with a couple of sample reductions.

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