Formal Specification of Abstract Datatypes
Exercises 4+5 (June 7+28)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.uni-linz.ac.at

May 25, 2010

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 (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)

Exercise 5: Strict Adequacy of Specification

Take above initial specification of `Queue` (where `Elem` is replaced by a sort `Nat` with free constructors `0` $:\rightarrow$ `Nat`, `s` : `Nat` $\rightarrow$ `Nat` which can be assumed to strictly adequately specify the algebra of natural numbers). Show that this specification is strictly adequate with respect to the classical algebra of queues using the proof technique of characteristic term algebras.

1. Give a full definition of the algebra of queues with functions for the operations in `Queue`.
2. Define a characteristic term algebra for `Queue` and prove that it is indeed characteristic.
3. With the help of the characteristic term algebra, prove the strict adequacy of `Queue` with respect to the algebra of queues.

Hints: the domain *Queue* of the classical algebra of queues can be defined as

$$\begin{aligned} &\mathit{Queue} := \textstyle\bigcup_{n\in\mathbb{N}} Q_n \\ &Q_n := \mathbb{N}_n \rightarrow \mathbb{N} \end{aligned}$$

where $\mathbb{N}_n$ is the set of the first n natural numbers. In other words, Q_n is the set of finite sequences (of natural numbers) of length n and *Queue* is the set of all finite sequences. Consequently, for every sequence $q \in Q_n$ and position $i \in \mathbb{N}_n$, the term $q(i)$ denotes the element at position i of sequence q.

Based on this domain, one can define the classical *Queue* functions, e.g.

$$\begin{aligned} &\mathit{length} : \mathit{Queue} \rightarrow \mathbb{N} \\ &\mathit{length}(q) = \text{such } i \in \mathbb{N} : q \in \mathit{Queue}_i \\ &\mathit{head} : \mathit{Queue} \rightarrow \mathbb{N} \\ &\mathit{head}(q) := q(0) \end{aligned}$$

As for proving properties of *Queue*, in order to show that for some predicate P, the formula $\forall q \in \mathit{Queue} : P(q)$ is true, it suffices to show that $\forall n \in \mathbb{N} : \forall q \in Q_n : P(q)$ is true (which can be shown by induction on n).