SOEN331: Introduction to Formal Methods for Software Engineering
Assignment 1 on algebraic specifications

Instructor: Dr. O. Ormandjieva

February 14, 2017

1 General information

Date posted: 1 February, 2017.
Date due: 16 February by 17:00.
Weight: 5% of the overall grade.

1. The primitive constructor creates an empty linked list.
2. For an empty linked list, `head` is null.
3. For an empty linked list, `tail` is null.
4. For a non-empty linked list, `head` points to the newly constructed node.
5. Adding an element to an empty list creates a linked list with only one element
6. Adding an element to a linked list will increase its size by 1.
7. There is nothing after the tail.
8. For a non-empty linked list, `tail` points to the last node.
9. For a newly added node, *data* holds the most recently added element.
10. Two nodes, containing elements $el2$ and $el1$, respectively, can be successfully linked according to the protocol of the ADT.

Solution:

Spec: Linked List (Element);
Sort: LinkedList;
Imports: Boolean, $\mathbb{N}$;
Description:

See problem statement.

Operations:

$create :\rightarrow LinkedList$;
$add : Element \times LinkedList \rightarrow LinkedList$;
$isEmpty : LinkedList \rightarrow Boolean$;
$getData : Node \rightarrow Element$;
$getNext : Node \rightarrow Node$;
$head : LinkedList \rightarrow Node$;
$tail : LinkedList \rightarrow Node$;

Variables:

$el1, el2 : Element; ll : LinkedList$

Axioms:

1. $isEmpty(create) = true$;
2. $head(create) = null$;
3. $tail(create) = null$;
4. $getData(head(add(el1, create))) = el1$;
5. $size(add(el1, create)) = 1$;
6. $size(add(el1, ll)) = size(ll) + 1$;
7. $getNext(tail(add(ell, create))) = null$;
8. $getData(tail(add(el2, add(el1, create))) = el1$;
9. $getData(head(add(el2, add(el1, create))) = el2$;
10. $getData(getNext(head(add(el2, add(el1, create))))) = el1$;