

SOEN331: Introduction to Formal Methods for Software Engineering

Assignment 1 on algebraic specifications

Instructor: C. Constantinides

February 13, 2017

Spec: `LinkedList(Node)`;

imports: `Whole (Natural with zero), Boolean`;

Sort: `LinkedList`;

Description: As mentioned in the assignment, "a Linked List is a collection of nodes. The collection is unordered and allows duplicates." Each node, of type `Node`, has two parts in the linked list: 1. `data` holds the element, of some generic type `Element`, stored in the current node, and 2. `next` holds a reference to the next node in the list.

Operations:

create : $\rightarrow \text{LinkedList}$;

add : $\text{Element} \times \text{LinkedList} \rightarrow \text{LinkedList}$;

isEmpty : $\text{LinkedList} \rightarrow \text{Boolean}$;

getData : $\text{Node} \rightarrow \text{Element}$;

getNext : $\text{Node} \rightarrow \text{Node}$;

head : $\text{LinkedList} \rightarrow \text{Node}$;

tail : $\text{LinkedList} \rightarrow \text{Node}$;

size : $\text{LinkedList} \rightarrow \mathbb{N}_0$;

Variables:

el1, el2 : `Element`; *n1, n2* : `Node`;

Axioms:

1. $\text{isEmpty}(\text{create}) = \text{true}$;
2. $\text{head}(\text{LinkedList}) = \text{null}$ if $\text{isEmpty}(\text{LinkedList}) = \text{true}$;
3. $\text{tail}(\text{LinkedList}) = \text{null}$ if $\text{isEmpty}(\text{LinkedList}) = \text{true}$;
4. $\text{head}(\text{LinkedList}) = \text{Node}$ if $\text{isEmpty}(\text{LinkedList}) = \text{false}$;
5. $\text{size}(\text{create}(\text{Node})) = 1$;
6. $\text{length}(\text{add}(\text{Node}, \text{LinkedList})) = \text{length}(\text{LinkedList}) + 1$;
7. $\text{getNext}(\text{tail}(\text{LinkedList})) = \text{null}$;

8. $getData(tail(LinkedList)) = element$ if $isEmpty(LinkedList) = false$;
9. $getData(Node) = element$ if $getData(head(create(Node))) = element$;
10. $getNext(n1) = n2$ if $[getData(n1) = el2 \wedge getData(n2) = el1]$;