

# SOEN331: Introduction to Formal Methods for Software Engineering

## Assignment 1 on algebraic specifications

Instructor: Dr. O. Ormandjieva

February 1, 2017

### 1 General information

**Date posted:** 1 February, 2017.

**Date due:** 16 February by 17:00.

**Weight:** 5% of the overall grade.

### 2 Introduction

This assignment can be done individually or in pairs and it must be prepared by using L<sup>A</sup>T<sub>E</sub>X.

### 3 Simplified Linked List ADT

A Linked List is a collection of nodes. The collection is unordered and allows duplicates.

Consider a linked list where each node, of type *Node*, has two parts:

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.

Let us refer to the sort as *LinkedList*. An informal description of the operations of the *LinkedList* is given below:

- *create* creates an empty linked list.
- *add(Element, LinkedList)* inserts a new node before the current first node in the *LinkedList* with a single element, holding data of type *Element* and where the second parameter would point to the next node (or `null` if this is used as a non-default constructor for the list). This operation returns the modified *LinkedList*.
- *isEmpty(LinkedList)* returns true if the *LinkedList* is empty; it returns false otherwise.

- *getData(Node)* returns *Element* stored in *Node*.
- *getNext(Node)* returns the next node after *Node* (or *null* if there is nothing after the current node).
- *head(LinkedList)* returns the first node of the *LinkedList*; *null* if the list is empty.
- *tail(LinkedList)* returns the last node of the *LinkedList*; *null* if the list is empty.
- *size(LinkedList)* returns the number of nodes in the *LinkedList*.

There are 10 axioms in this specification. They correspond to the following rules:

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. A non-default constructor creates a linked list with only one element
6. Adding a new node 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 constructed node, *data* holds the most recent element provided to the non-default constructor.
10. Two nodes, containing elements *el2* and *el1*, respectively, can be successfully linked according to the protocol of the ADT.

**Number the axioms in your specification to correspond to the above rules.**

## 4 What to submit

You must produce the specification using the L<sup>A</sup>T<sub>E</sub>Xtext formatting package, and name it after your Concordia id, e.g. 123456.pdf (or a concatenation of your id's). Submit your pdf file under **Theory Assignment 1**. In the case of joint work, only one of you would need to submit.

## 5 Late submissions

Any late submission within the first 24 hours will get a 50% penalty and it will subsequently receive a 10% penalty per day.