

# SOEN331: Introduction to Formal Methods for Software Engineering

## Assignment 4 on algebraic specifications

Tarek Ait Hamouda (40044119), Abhijit Gupta (40066502),  
Ethel Narra Pangan (40061530)

April 6, 2019

**Spec:** Location;

**Sort:** Location;

**Imports:**String, Point

**Description:** A location contains a description

**Operations:**

newlocation:  $\text{String} \times \text{Point} \rightarrow \text{Location}$ ;

setdescription:  $\text{String} \times \text{Location} \rightarrow \text{Location}$ ;

getdescription :  $\text{Location} \rightarrow \text{String}$ ;

setpoint :  $\text{Point} \times \text{Location} \rightarrow \text{Location}$ ;

getpoint :  $\text{Location} \rightarrow \text{Point}$ ;

**Variables:**

newDesc, d: String; newPoint, p: Point;

**Axioms:**

[A1]  $\text{getdescription}(\text{newlocation}(d, p)) = d$ ;

[A2]  $\text{getpoint}(\text{newlocation}(d, p)) = p$ ;

[A3]  $\text{setdescription}(\text{newDesc}, \text{newlocation}(d, p)) = \text{newlocation}(\text{newDesc}, p)$ ;

[A4]  $\text{setpoint}(\text{newPoint}, \text{newlocation}(d, p)) = \text{newlocation}(d, \text{newPoint})$ ;

**Spec:** Map (Location);

**Sort:** Map;

**Imports:** String, Point, Boolean, Location,  $\mathbb{N}$ ;

**Description:** A map contains locations

**Operations:**

newmap:  $\rightarrow$  Map;  
addlocation:  $\text{Map} \times \text{Location} \rightarrow \text{Map}$ ;  
deletelocation :  $\text{Map} \times \text{String} \rightarrow \text{Map}$ ;  
containsdescription :  $\text{Map} \times \text{String} \rightarrow \text{Boolean}$ ;  
containspoint :  $\text{Map} \times \text{Point} \rightarrow \text{Boolean}$ ;  
findlocation :  $\text{Map} \times \text{String} \rightarrow \text{Point}$ ;  
isempty :  $\text{Map} \rightarrow \text{Boolean}$ ;  
clear :  $\text{Map} \rightarrow \text{Map}$ ;  
size :  $\text{Map} \rightarrow \mathbb{N}$

**Variables:**

d: String; p, q: Point; loc: Location; map: Map

**Axioms:**

[A1] isempty(newmap) = true;  
[A2] isempty(clear(map)) = true;  
[A3] containsdescription(addlocation(map, new), getdescription(loc)) = true;  
[A4] containsdescription(map, d)  $\rightarrow$  findlocation(addlocation(map, newlocation(d, q)), d)  
== q

[A5] size (addLocation(map, newlocation(d, q))) = **if** (containsdescription(map, d)) **then**  
size(map) **else** size(map) + 1

[A6] isempty(deleteLocation(addlocation(newmap, newlocation(d, p)), d)) = true

[A7] findlocation(addlocation(map, newlocation(d, p)), d) = p

[A8] findlocation(newmap, d) = undefined;

[A9] deletelocation(newmap, d) = undefined;

**preconditions:**

```
pre : deletelocation(map: Map, d: String) = containsdescription (map, d);  
pre : findlocation(map: Map, d: String) = containsdescription (map, d);
```