272 – Homework Assignment 1

Fall 2018

Due: Thursday, October 25th, 11:00AM in class

Do not discuss the problems with anyone other than the instructor.

Instructions: You need to install and run Alloy Analyzer for problems 2 and 3. Turn in a hard copy of your answers at the beginning of class on the due date. Additionally, for problems 2 and 3, send an ASCII text file via email to bultan@cs.ucsb.edu (with subject line “272 HW1”) containing your Alloy specifications and the outputs generated by the Alloy Analyzer.

1. Computer science program is made up of **classes** and two types of people **professors** and **students**. There are two types of students: **graduate students** and **undergraduate students**.

Classes are taught by professors. Undergraduate classes are taken by only undergraduate students and can have at most 50 students. Graduate classes can be taken by both undergraduate and graduate students and can have at most 30 students.

Professors can advise up to ten graduate students. Each graduate student can have at most one professor as an advisor. Each professor has a research budget. Hiring a graduate student costs a professor $50K out of his research budget. Professors can only advise graduate students if they have the research budget available to fund them.

Graduate students have to do directed research for a professor before they can agree on an advisor/advisee relationship. Graduate students should not do directed research for a professor who does not have the research budget available to support a graduate student. A professor may have unlimited number of grad students doing directed research at one time, provided that they have the funds to hire at least one grad student.

Assume that we have the following operations, **hireGrad**, **releaseGrad**, **beginDirectedResearch**, **takeClass**. hireGrad is an operation where a Professor decides to hire a grad student to work for them and becomes an advisor for that grad student. The research budget of the professor is updated when they hire a graduate student. releaseGrad occurs when a professor's grad student graduates the program or no longer works for the professor. beginDirectedResearch occurs when a grad student starts a research project with a professor. takeClass occurs when a student enrolls to a class. These operations have to satisfy the constraint stated above.

Write UML class diagrams and OCL constraints for modeling the above description. Include the multiplicities for the associations in your class diagrams. Write class invariants and pre- and post-conditions of the operations based on the above description.

2. Consider a **Doubly Linked List (DLL)**. If a DLL is not empty, its **head** is a **node** that does not have a **prev** node. If a DLL is not empty, its **tail** is a node that does not have a **next** node. An empty DLL does not have a head node. The **contents** of a DLL is the set of nodes that are reachable from the head node by following the next links, plus the head node. The tail node is included in the contents. For all nodes that are reachable from the head node, the next of the prev of the node is itself.

Write the DLL specification in Alloy. Check or simulate the following properties for scopes 2 and 3 using the Alloy Analyzer: 1) There exist DLLs with 0, 1, 2, and 3 nodes. 2) A DLL does not have a tail node if and only if it is empty. 3) For all DLLs, if head and tail are the same node, then the size of the DLL is 1. 4) No node in a DLL is the prev of two different nodes or the next of two different nodes. 5) There are no cycles (i.e., a node is not reachable from itself by just following next links or by just following prev links).

Turnin printout of the Alloy specifications and the ouput of the Alloy Analyzer.

3. Extend the above DLL specification in Alloy by writing the predicates for the **add** and **delete** operations. Assume that the add operation adds the new node to the head of the DLL (i.e., the new node becomes the new head) and the delete operation removes the tail node (i.e., the prev of the tail becomes the new tail).

Hint: Specify prev and next as relations (using the cross product `->`).

Check or simulate the following properties for scopes 2 and 3 using the Alloy Analyzer: 1) It is possible to obtain an empty DLL after a delete. 2) After an add the DLL contains at least one node. 3) Adding a node to a DLL increases the size of its contents by one. 4) Deleting a node from a DLL decreases the size of its contents by one. 5) After an add, the next of the new head is the old head. 6) After a delete, the new tail is the prev of the old tail.

Turnin printout of the Alloy specifications and the ouput of the Alloy Analyzer.

4. Given the following Alloy formula:

```
r: X ->one Y
all x:X | x != x.r.~r
```

Assume that the scope for X and Y are 2 and generate a SAT formula which is satisfiable only if the above formula is not valid (i.e., negation of the above formula is satisfiable) based on the approach presented in the lecture notes on Alloy Analyzer.