CS1760: Multiprocessor Synchronization Due 16 November 2023

Project 2: DupLists

Professor: Maurice Herlihy *TA: cs1760tas@lists.brown.edu*

1 Introduction

In this assignment you will extend the OptimisticList, LazyList, and LockFreeList list algorithms described in the lecture and textbook so that they represent multisets, which are lists that can contain duplicates.

2 Requirements

Your list implementations will follow the following specifications:

- add(T item): should increase the number of instances of of the item in the list by one. Then return True
- remove(T item): should return False if no instances of item appear in the list. Otherwise it should decrease the number of instances of the item in the list by one and return True
- contains(T item) should return True when there is 1 or more instances of item in the list. It should return False otherwise
- you do not need to count the number of instances of an item in your list
- you might also need to fill out some helper functions or create your own. Places where code is needed is marked by TODO

Explain whether each of the functions mentioned above for each implemented list is lock-free, wait-free, or neither in 1-2 sentences in a README file. Ensure that your answers are consistent with the corresponding implementations discussed in the lecture.

3 Stencil

You can find the stencil code HERE. Once you have the stencil code the files you need to modify are:

- LazyDupList.java
- LockFreeDupList.java
- OptimisticDupList.java

You might need to read the interface files for understanding of how some base implementation works. For example in lazy and optimistic you can see that nodes have locks.

4 Testing

For this assignment, you are not required to write your own test cases; however, you are more than welcomed to do so. You can run the provided tests with **make run1**, **make run10**, **make run100**, and **make run1000** commands.

Notice that passing any test cannot guarantee your implementation to be correct, so we will manually check the correctness of your code during grading.

5 Tips

Here are some tips to get started:

- Most of the code you will need for this assignment will be from the book so feel free to use it!
- You must use the stencil code in your implementation
- If you are seeing null when running the provided tests, it means you have a bug somewhere.