Practice 3:

Harsh Mohan Sason

2. Solution 3 for the too much milk problem is definitely safe because in this solution at least one of the threads determines whether the other thread has bought milk or not before deciding whether or not to buy. We can use the same argument to prove this Solution 3 is safe using the similar argument used in Solution to which was considering three cases:

Case1: (noteB = 1, milk = any value). It contradicts the assumption that thread A buys milk and reaches A3.

Case2: (noteB = 0, milk > 0). The property milk> 0 is a stable property, i.e once it is true, it remains true thus contradicting our assumption

Case3: (noteB = 0, milk = 0). This case shows us that B cannot buy milk in the future (either the test B1 or B2 must fail), which contradicts our assumption that both A and B buy milk.

Thus every case contradicts the assumption, proving the algorithm is safe.

3. The set of possible outputs that could occur are:

a.) All removals may happen after the insertions, thus 0-9 will be removed

b.) Removals may or could happen before the insertions in which case 0-9 will be removed.

c.) Interleaving can happen thus causing unto 30 items being inserted into the queue and only 20 being removed

d.) Because of the join, insertion will put 0-9 in the queue, and other insertions will have no effect and same is the case with when the main returns from the join.

e.) Items which are removed could not necessarily be sequential

10. In the example, we use writeGo.Signal rather than writeGo.Broadcast because when a read is finished or complete, at most one write can progress. Thus using writeGo. Broadcast is not at all needed and is unnecessary.