Software testing homework

name：徐政辉 class：17计科全英文 number：2017329621139

**question**

1. **Use the class Queue in Figure 2.40 for questions (a)–(f) below. The queue is managed in the usual circular fashion.**

**Suppose we build a FSM where states are defined by the representation variables of Queue. That is, a state is a 4-tuple defined by the values for [elements,size, front, back]. For example, the initial state has the value [[null, null], 0, 0, 0], and the state that results from pushing an object obj onto the queue in its initial state is [[obj, null], 1, 0, 1].**

1. **We do not actually care which specific objects are in the queue. Consequently, there are really just four useful values for the variable elements.Enumerate them.**

[null, null], [obj, null], [null, obj], [obj, obj]

1. **How many states are there?**

[[null, null], 0, 0, 0] √

[[null, null], 0, 1, 1] √

[[obj, null], 1, 0, 1] √

[[obj, null], 1, 1, 0]

[[obj, obj], 2, 1, 1] √

[[obj, obj], 2, 0, 0] √

[[null, obj], 1, 0, 1]

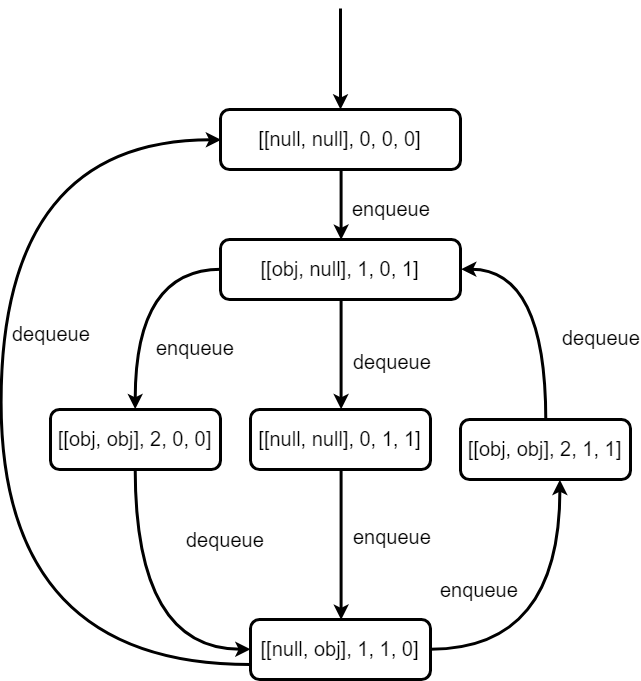
[[null, obj], 1, 1, 0] √

There are eight states

1. **How many of these states are reachable?**

Only six states are reachable. As showed in problem (b) which I use “√” to mark.

1. **Show the reachable states in a drawing.**



1. **Add edges for the enqueue() and dequeue() methods. (For this assignment, ignore the exceptional returns, although you should observe that when exceptional returns are taken, none of the instance variables are modified.)**

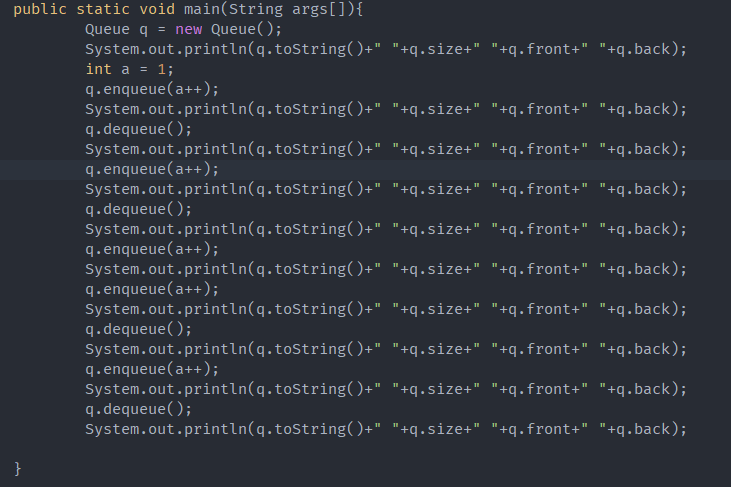
The graph is just showed above as problem (d).

1. **Define a small test set that achieves edge coverage. Implement and execute this test set. You might find it helpful to write a method that shows the internal variables at each call.**

Test set:

Enqueue->dequeue->enqueue->dequeue->enqueue->enqueue->dequeue->enqueue->dequeue

Test code:



Output:

