Sri Hari Malla - CS19BTECH11039
Prof : Dr. Sathya Peri
Parallel and Concurrent Programming
Indian Institute of Technology Hyderabad

THEORY ASSIGNMENT - II

Sri Hari Malla - CS19BTECH11039

February 10, 2022

Alteration of the m-valued Regular MRSW register:

Given modified code:

```
public class RegularMRSWRegister implements Register<Byte> {
    private static int RANGE = Byte.MAX_VALUE - Byte.MIN_VALUE + 1;
    boolean[] r_bit = new boolean[RANGE]; // regular Boolean MRSW
    public RegularMRSWRegister(int capacity) {
        for (int i = 1; i < r_bit.length; i++) {
            r_bit[i] = false;
        }
        r_bit[0] = true;
    }
    public void write(Byte x) {
        for (int i =0; i <= x - 1; i++) {
            r_bit[i] = false;
        }
        r_bit[x] = true;
    }
    public Byte read() {
        for (int i = 0; i < RANGE; i++) {
            if (r_bit[i]) {
                return i;
            }
        }
        return -1; // impossible
    }
}
```

Initially when the class is instantiated, there are no readers, and the constructor sets all bits to false except the first bit. For any read, let x be the value written by the most recent non-overlapping write. At the time the write operation is completed, r_bit[x] was set to true and r_bit[i] is false for every i < x. By lemma 4.2.3 from book if the reader returns a value that is not x, then it is observed that some bit r_bit[j], `j != x` to be true and that bit must have been set by a concurrent write.

Case: Consider a case where after some multiple operations of read and write on the register, in which the read is at 4 and write is at 5.

In the original code given, without loss of generality let's assume that the write operation by some thread `w` writes the r_bit[5] to true first and then set all the subsequent i < 5 to false. While reader reads it finds the position at 5 and return it.

```
0 1 2 3 4 5 6 7
- - - - r w - - (r ->)
- - f f f T - -
- - - - - r - -  (reader returns 5)
```

But here in this case, in the same situation, when writer thread initiates the function it sets the subsequent i < 5 to false and then set the 5th bit to true, meanwhile in which the reader might pass the 5th bit and returns -1 as it reaches to the end of the bit.

```
0 1 2 3 4 5 6 7
- - - - r w - - (r ->)
f f - - - r - -
f f f f f t - r (reader returns -1)
```

As discussed, a regular MRSW registers returns a value corresponding to the bit 0 to M-1 set by some write call. But here there is a problem that the reader might return `-1` violating the lemma 4.2.3 of the book.

Even though it seems as a regularized MRSW register, clearly by this contradiction, the modified register doesn't always exhibit regularization property.

LaTeX generated document

THE END