Theory Assignment -2

Souvik Sarkar
(CS23MTECH02001)

Exercise 4.2

Consider the safe Boolean MRSW construction shown in Fig. 4.6. True or false: If we replace the safe Boolean SRSW register array with an array of regular Boolean SRSW registers, then the construction yields a regular Boolean MRSW register. Justify your answer.

Ans:

true:If we replace the safe Boolean SRSW register array with an array of regular Boolean SRSW registers, then the construction does yield a regular Boolean MRSW register.

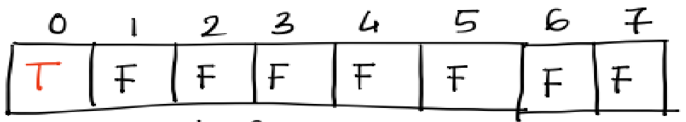
The proof is almost exactly the same.For non-overlapping method calls, each table[i] holds the most recently written value, which is returned by the read() call. For overlapping method calls, the reader may return either the new value or the old value, because the component registers are regular.

Exercise 4.7

You are given the algorithm in Fig. 4.22 for constructing an atomic M-valued SRSW register using atomic Boolean SRSW registers. Does this proposal work? Either prove the correctness or present a counterexample.

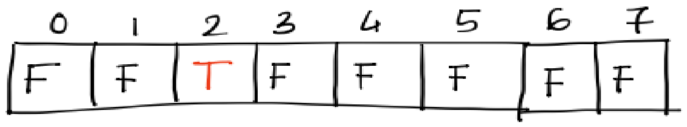
```java
public class AtomicSRSWRegister implements Register<int> {
  private static int RANGE = M;
  boolean[] r_bit = new boolean[RANGE]; // atomic boolean SRSW
  public AtomicSRSWRegister(int capacity) {
    for (int i = 1; i <= RANGE; i++)
      r_bit[i] = false;
    r_bit[0] = true;
  }
  public void write(int x) {
    r_bit[x] = true;
    for (int i = x - 1; i >= 0; i--)
      r_bit[i] = false;
  }
  public int read() {
    for (int i = 0; i <= RANGE; i++)
      if (r_bit[i]) {
        return i;
      }
    return -1; // impossible
  }
}
```

Ans:

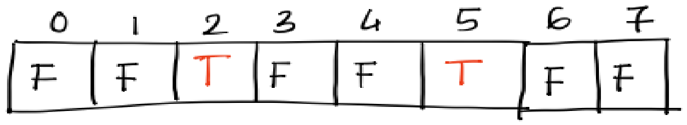


Atomic Boolean SRSW array

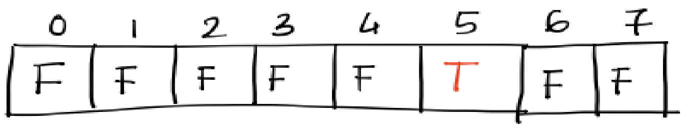
Case 1:

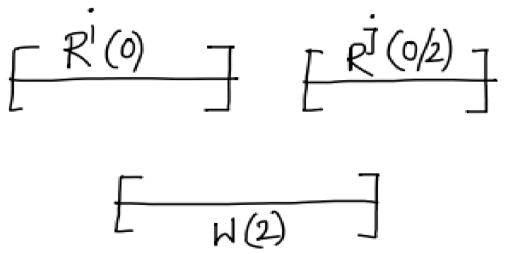


Case 2:



Case 3:





If M is the total number of potential values, let's say that we wish to build an atomic M-valued SRSW register. This can be done by employing a collection of atomic Boolean SRSW registers. The writer thread sets a flag (Boolean) in each entry of the array, which corresponds to one of the available values, to indicate the chosen value. Every entry in the array is initially set to "false," with the exception of the entry at index 0, which is set to "true." When writer thread wants to write a new value at a certain index, it sets the entry for that index to "true," and all other indexes below that index are set to False. When writer thread reads, it returns the index of the first "true" item as the selected value, which is 4, when there are multiple "true" entries.

- Case 1: The reader thread in this instance is Thread 1. The value at index zero, which is initially read, is set to "True." If writer thread updates the value at index 2, Thread 1 can choose between two options:
    - Thread 1 will return the value 2 (the freshly written value) if writer thread is quick and writes to index 2 before Thread 1 reads.
    - In the event that writer thread is slower and Thread 1 reads index 2 before writing to it, Thread 1 will return the value 0 (the most recent value written).
    - Because Thread 1 returns either the most recently written value or the most recent value, depending on the relative time of the read and write operations, this situation satisfies the requirement for an atomic register.
- Case 2 and 3

    Let's now think about a situation in which writer thread writes the value at index 5.Comparable to Case 1, Thread 1 has two options:
        - The value 5 (the freshly written value) will be returned if writer thread writes to index 5 before Thread 1 reads it.
        - If writer thread is slower and Thread 1 reads index 5 before writer thread writes it, Thread 1 will return value 2, which is the most recent value written.

Once more, in this instance, the need of an atomic register is satisfied since Thread 1 either returns the recently written value or the most recent written value depending on the time of the operations.
The use of atomic Boolean SRSW registers in both scenarios assures that Thread 1 receives a consistent and valid value, either the recently written value or the most recent written value, even when Thread 1 is not the only register in use. Thread 1 and writer thread, They operate in parallel with writer thread. Using atomic Boolean SRSW registers and an M-valued SRSW register, where M is the total number of potential values, this architecture enables the building of an atomic SRSW register.