CS5300 Theory Assignment 2

Gautam Singh
CS21BTECH11018

1) **True.** If thread A's `read()` call does not overlap any `write()` call to its location, then this call will return the most recently written value into `s_table[`A`]`. If it does, then it will return the new value or the old value since the component registers are regular. Thus, the construction yields a regular Boolean MRSW register.
2) **True.** Since the SRSW registers are regular, no read call can return a value from either the future or the distant past. We only need to check if an earlier read can return a value later than that returned by a later read. Since the SRSW registers are regular, this can only happen when reads overlap a single write.
   Suppose an earlier read returns the old value. Then, the later read can return either the old or new value, depending on whether the diagonal entry for that thread has been updated by the writer thread.
   Otherwise, if the earlier read returned the new value, it would have also updated all the values in its column to reflect the newly read value. A later read would then scan its row to find this updated value written by the earlier reader (if not written by the writer thread) and thus return the same value as it has the highest timestamp.
3) Obviously, the reader thread cannot read values from the future since the entry in `r_bit` will not be set. Suppose $W^i \rightarrow W^j$. If the latter value is smaller, then any read following it will read this value. If it is larger, then the earlier read value is wiped out, and a later read will still read the newer value.
   Now, we need to prove that a later read returns a value later than the earlier read. If both reads do not overlap with a write, then the later read cannot contain an earlier value. If they overlap with a write and the earlier read reads the new value, the later read returns the same value since the array of registers is atomic, thus the same boolean value at the same array location will be read by the later read.
4) Clearly, this construction guarantees a safe register, since the returned value is also a 64-bit value. However, if a read operation overlaps with a write operation, then it can read the new upper 32 bits and the old lower 32 bits. This is neither characteristic of regular or safe registers. Thus, the strongest property satisfied is a safe register.