ABA PROBLEM

When a memory address is read twice, and it has the same value for both reads; it is often used to interpret that "nothing has changed". This is usually the case with lock-free algorithms, where the programmer uses atomic instructions like compare-and-set (CAS) which compares old-value and new-value, and if they match it updates the value.

However, it is possible that another thread executed in between the two reads and changed the original value “A” to a new value “B”, did some more work, and finally change the value back to “A”. The first thread is thus fooled into thinking that "nothing has changed" even though the second thread did work that violates that assumption.

Such a problem can occur with garbage collection when a “freed” node is reused in exactly the same place where it was originally present, but its “next” nodes are not the same. Considering this “new” node to be the same as the original node is the ABA problem.