(b)

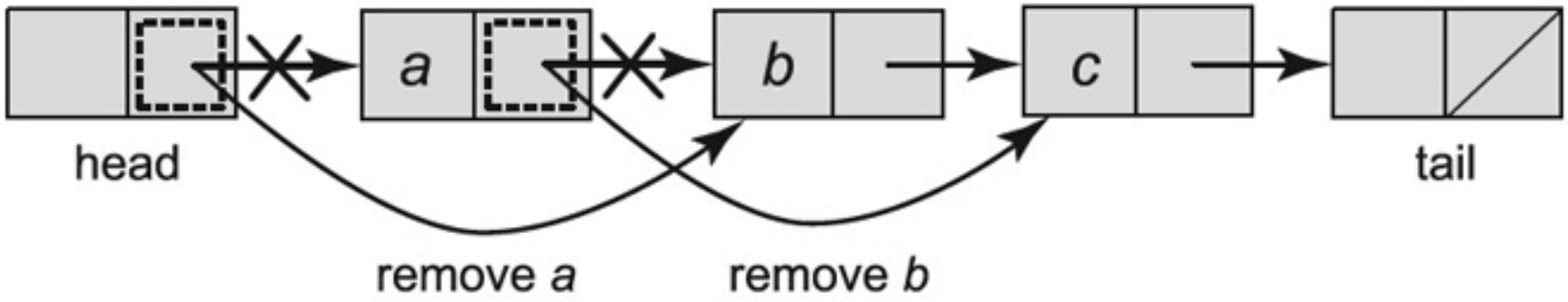

FIGURE 9.21B The LockFreeList class: why mark and reference fields must be modified atomically. In part (a), thread *A* is about to remove *a*, the first node in the list, while *B* is about to add *b*. Suppose *A* applies compareAndSet() to head.next, while *B* applies compareAndSet() to a.next. The net effect is that *a* is correctly deleted but *b* is not added to the list. In part (b), thread *A* is about to remove *a*, the first node in the list, while *B* is about to remove *b*, where *a* points to *b*. Suppose *A* applies compareAndSet() to head.next, while *B* applies compareAndSet() to a.next. The net effect is to remove *a*, but not *b*.