



FIGURE 9.19A The LazyList class: why validation is needed. In part (a), thread A is attempting to remove node a . After it reaches the point where pred_A refers to curr_A , and before it acquires locks on these nodes, the node pred_A is logically and physically removed. After A acquires the locks, validation will detect the problem. In part (b), A is attempting to remove node a . After it reaches the point where pred_A refers to curr_A , and before it acquires locks on these nodes, a new node is added between pred_A and curr_A . After A acquires the locks, even though neither pred_A nor curr_A is marked, validation detects that pred_A is not the same as curr_A , and A's call to `remove()` will be restarted.