retire[] is logical delete. retire(ptr_k_A) means ptr_k_A is not being held by any cpk_q. when ptr_k_A is no longer hazardous or being protected, it will be deleted. another thread can still reference ptr_k_A to get k_A but $ptr\_k_A \notin$ any $cpk_i \Rightarrow k_A \notin dB$ so returning v_A would be a correctness issue.

$s_i == F$ ✓

$maybe\_ptr\_k_i := cpk_i.load()$

$maybe\_ptr\_k_i \neq \phi$ ✓

$reload := cpk_i.load()$

$reload == maybe\_ptr\_k_i$ ✓

$ptr\_k_i := maybe\_ptr\_k_i$

protect()

$ptr\_k_i \neq \phi$ ✓

* $ptr\_k_i == k_x$ ✓

$ptr\_k_i \neq \phi$

$cas(F \rightleftharpoons X)$ // no deletes of k_x are possible now

$ptr\_k_i \neq \phi$

retire($ptr\_k_i \cap ptr\_v_i$)

ptr_k_i could have been swapped here for ptr_k_i' but this fine

ptr_k_i swapped with $ptr\_k_i' \rightarrow k_x'$: $ptr\_k_i \neq \phi$

$cas[F \rightleftharpoons X]$ // no deletes of k_x are possible now

swap results in this branch

last delete results in this branch

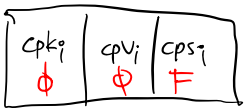
$ptr\_k_i \neq \phi$ ✓

k_x' deleted instead of k_x!

$ptr\_k_i \rightarrow \phi$ ✓

$ptr\_k_i \neq cpk_i.load()$ ✓

$cps_i := F$



currupted state!