c.orden = subseq(co.orden, 1, capacidad) ++ [e]) ∧L

(e ∉ co.dicc ∧ |co.orden| > co.capacidad ⟶

c.dicc = setkey(co.dicc, e, v) ∧ c.orden = co.orden ++ [e])

}

b

TAD LRU<K,V> {

obs dicc : dicc<K,V>

obs cap : N

obs accesos : <K, Z>

proc nuevo(in c: N): LRU<K,V> {

asegura {res.dicc = {} ∧L res.cap = c ∧L res.accesos = {}}

}

proc esta(in c: LRU<K,V>, in e: K): bool {

asegura {e ∈ c.dicc}

}

proc obtener(in c: LRU<K,V>, in e: K): V {

requiere {e ∈ c.dicc}

asegura {res = c.dicc[e]}

}

proc definir(inout c: LRU<K,V>, in e: K, in v: V)

Requiere {c = c0}

asegura {

((e ∈ c0.dicc) v (e ∉ c0.dicc ∧ cap > |c0.dicc|))

c.dicc = setkey(c0.dicc, e, v) ∧

c.accesos = setkey(c0.accesos, e, horaActual()) ∧L

18 (R) Si, reemplaza

16 (P) setkey reemplaza una key y o definida o antes hay que borrarla?