```
1  {V ⊩ p ↦_m _ * F_tid ∧ [H * ∃y. C ↦ y * y ↦ _ * true] ∧ [p ↦_e _ * true]}
2  void retire(int* p) {
3    {V ⊩ p ↦_m _ * ∃A. detached ↦ A * D(A) ∧ [H * ∃y. C ↦ y * y ↦ _ * true] ∧ [p ↦_e _ * true]}
4  insert(detached,p);
5    {V ⊩ ∃A. detached ↦ A * D(A) ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
6    if (nondet())
7      {V ⊩ F_tid ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
8      return;
9    Set used;
10   {V ⊩ ∃A. detached ↦ A * D(A) ∧ used = ∅ ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
11   while (!isEmpty(detached)) {
12     {V ⊩ ∃A. detached ↦ A * D(A) * D(used) ∧ A ≠ ∅ ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
13     bool my = true;
14     Node *n = pop(detached);
15     {V ⊩ my ∧ ∃A. detached ↦ A * D(A) * D(used) * n ↦_m _ * [n ↦_e _ * true] * [H * ∃y. y ≠ n ∧ C ↦ y * y ↦ _ * true]}
16     for (int i = 0; i < N && my; i++) {
17       {V ⊩ my ∧ 0 ≤ i < N ∧ ∃A. detached ↦ A * D(A) * D(used) * n ↦_m _ * [n ↦_e _ * true] *
18         [H * ∃y. y ≠ n ∧ C ↦ y * y ↦ _ * true] ∧ ∀0 ≤ j < i. [∃y. y ≠ n ∧ C ↦ y * y ↦ _ * true] since [HP[i] ≠ n * true]}
19       if (⟨HP[i] == n⟩_ld)
20         my = false;
21       {V ⊩ 0 ≤ i < N ∧ ∃A. detached ↦ A * D(A) * D(used) * n ↦_m _ * [n ↦_e _ * true] *
22         [H * ∃y. y ≠ n ∧ C ↦ y * y ↦ _ * true] ∧ (my ⇒ ∀0 ≤ j ≤ i. [∃y. y ≠ n ∧ C ↦ y * y ↦ _ * true] since [HP[j] ≠ n * true])}
23     }
24     {V ⊩ ∃A. detached ↦ A * D(A) * D(used) * n ↦_m _ * [n ↦_e _ * true] ∧
25       [H * ∃y. C ↦ y * y ↦ _ * true] ∧ (my ⇒ ∀0 ≤ j ≤ N. [∃y. y ≠ n ∧ C ↦ y * y ↦ _ * true] since [HP[j] ≠ n * true])}
26     if (my) {
27       ⟨ ; ⟩_Take;
28       {V ⊩ ∃A. detached ↦ A * D(A) * D(used) * n ↦ _ * [H * ∃y. C ↦ y * y ↦ _ * true]}
29       free(n);
30     } else {
31       insert(used, n);
32     }
33     {V ⊩ ∃A. detached ↦ A * D(A) * D(used) ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
34   }
35   {V ⊩ detached ↦ ∅ * D(used) ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
36   moveAll(detached, used);
37   {V ⊩ F_tid ∧ [H * ∃y. C ↦ y * y ↦ _ * true]}
38 }
```