

Input: Variable *var* at code in function *func*

Output: Post-deallocation macro

```
1: Variables
2:   CopyAnalyser: Copy elimination analyser
3:   LiveAnalyser: Live variable analyser
4: end Variables
5: procedure CHOOSEPOSTMACRO(var, code, func)
6:   macros := [] // Store all macros used in code
7:   if code is ArrayGenerator then
8:     target = array variable of code
9:     macros.append(NEWARR_DEALLOC_POST(target))
10:  else if code is Assignment then
11:    lhs = left-handed side of code
12:    rhs = right-handed side of code
13:    // Check if copy of rhs is removed at code
14:    if CopyAnalyser.isCopyRemoved(rhs, code, func) then
15:      macros.append(TRANSFER_DEALLOC_POST(lhs, rhs))
16:    else
17:      macros.append(ADD_DEALLOC_POST(lhs, rhs))
18:    end if
19:  else if code is Function call then
20:    lhs = target variable of code
21:    callee = called function of code
22:    for each aParam in code do // Actual Parameter aParam
23:      if aParam is an Array then // Macro is for array type only
24:        // aParam is live after code at func?
25:        isLive  $\leftarrow$  LiveAnalyser.isLive(aParam, func)
26:        fParam = map aParam at caller func to
27:          formal parameter at called function callee
28:        // fParam is mutated or returned by callee?
29:        isMutated  $\leftarrow$  MutateAnalyser.isMutated(fParam, callee)
30:        isReturned  $\leftarrow$  ReturnAnalyser.isReturn(fParam, callee)
31:        // Choose post macro based on above analysis results
32:        switch isMutated – isReturned – isLive do
33:          case  $F - F - T \vee F - F - F \vee T - F - F$ 
34:            macros.append(RETAIN_DEALLOC_POST(lhs, arg))
35:          case  $F - T - F \vee T - T - F$ 
36:            macros.append(RESET_DEALLOC_POST(lhs, arg))
37:          case  $F - T - T \vee T - T - T$ 
38:            macros.append(CALLER_DEALLOC_POST(lhs, arg))
39:          case  $T - F - T$ 
40:            macros.append(CALLEE_DEALLOC_POST(lhs, arg))
41:          end if
42:        end for
43:      else // No needs to use post-deallocation macro
44:        end if
45:    end procedure
```