

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

Output: Return *true* if the copy of variable *var* is removed.

```
1: Variables
2:   MutateAnalyser: Read/Write Analyser
3:   ReturnAnalyser: Return analyser
4:   LiveAnalyser: Live variable analyser
5:   isLive: Is var still used/live after code in func
6:   isMutated: Is var mutated by called function callee
7:   isReturned: Is var returned by called function callee
8: end Variables
9: procedure ISCOPYREMOVED(var, code, func)
10:  if code is Assignment then
11:    // Check if var is used after code in func
12:    isLive  $\leftarrow$  LiveAnalyser.isLive(var, code, func)
13:    return  $\neg$ isLive // Remove copy when var is NOT live
14:  else if code is Function Call then
15:    callee = get called function from code
16:    param = map var to formal parameter of function callee
17:    // Check if param is mutated by function callee
18:    isMutated  $\leftarrow$  MutateAnalyser.isMutated(param, callee)
19:    // Check if param is returned by function callee
20:    isReturned  $\leftarrow$  ReturnAnalyser.isReturn(param, callee)
21:    isLive  $\leftarrow$  LiveAnalyser.isLive(var, code, func)
22:    if  $\neg$ isLive OR ( $\neg$ isMutated AND  $\neg$ isReturned) then
23:      return true // Eliminate the copy
24:    end if
25:    return false // Keep the copy
26:  else // No needs to optimise the code
27:    end if
28: end procedure
```