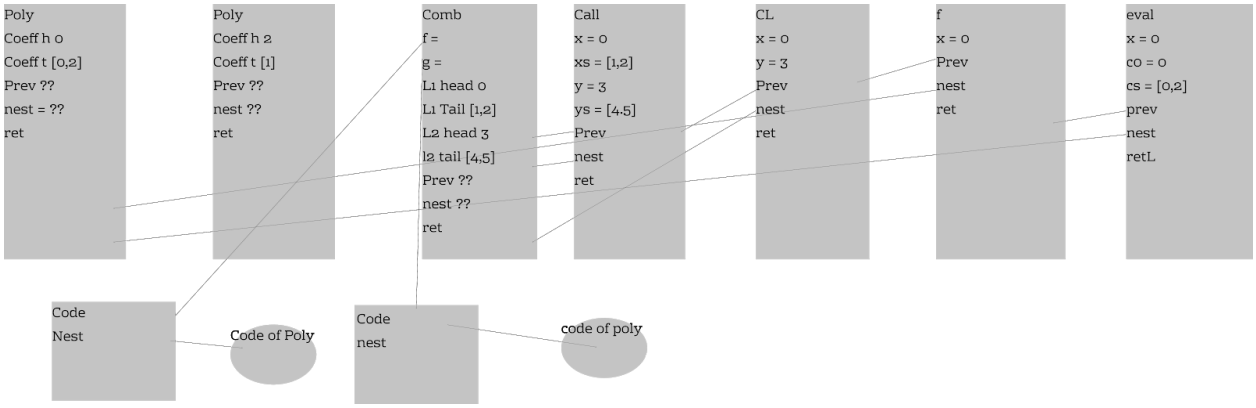
Poly
Coeff h 0
Coeff t [0,2]
Prev ??
nest = ??
ret
Poly
Coeff h 2
Coeff t [1]
Prev ??
nest ??
ret
Comb
f =
g =
L1 head 0
L1 Tail [1,2]
L2 head 3
l2 tail [4,5]
Prev ??
nest ??
ret
Call
x = 0
xs = [1,2]
y = 3
ys = [4,5]
Prev
nest
ret
CL
x = 0
y = 3
Prev
nest
ret
f
x = 0
Prev
nest
ret
eval
x = 0
c0 = 0
cs = [0,2]
prev
nest
retL
Code
Nest
Code of Poly
Code
nest
code of poly

The new activation link's nested link should be pointed to the scope of where the function is declared
m <n: it can be activated but recursively checking the nested links until it hits the targeted function in which they will create a new link to that function
m = n: Simply check the nth activation record for that specific function and call it and create a new link
m > n: this function can simple make the call to that function and simply execute it since the function is inside the memory the function allocates .
m > n + 1: is not possible because of how there is no active link to m and m is not in the current scope.

Yes Ruby uses Mark and Sweep as its garbage collector. It is build during language implementation.
Source:https://ruby-hacking-guide.github.io/gc.html