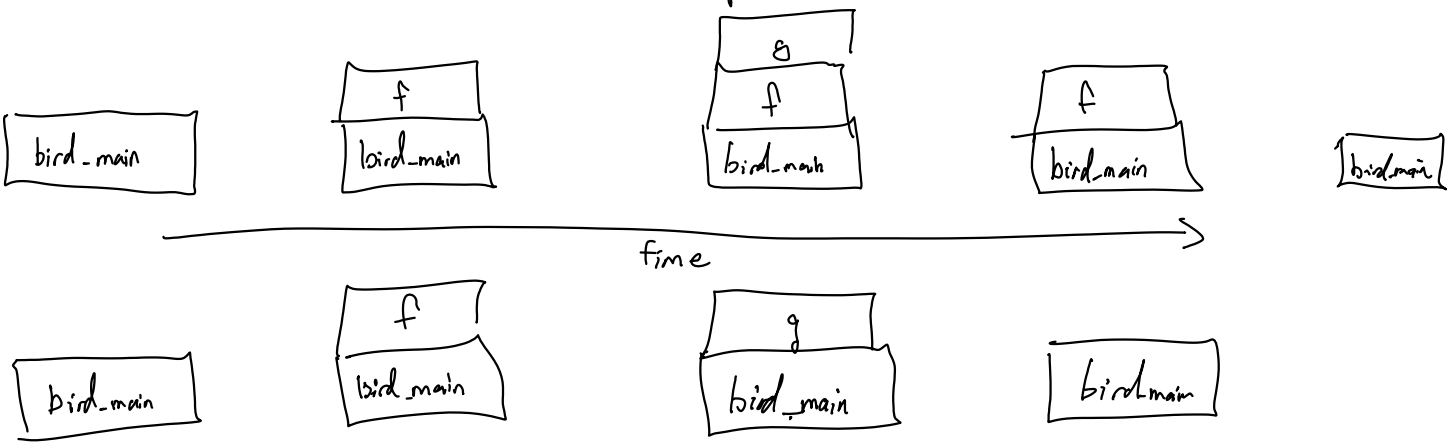
Tail-Call Optimization

What is? Tail Call: a call which is the last step of evaluating the body of the function that contains it

Tail Call Optimization (TCO): reclaim stack memory at a tail call site before making the call

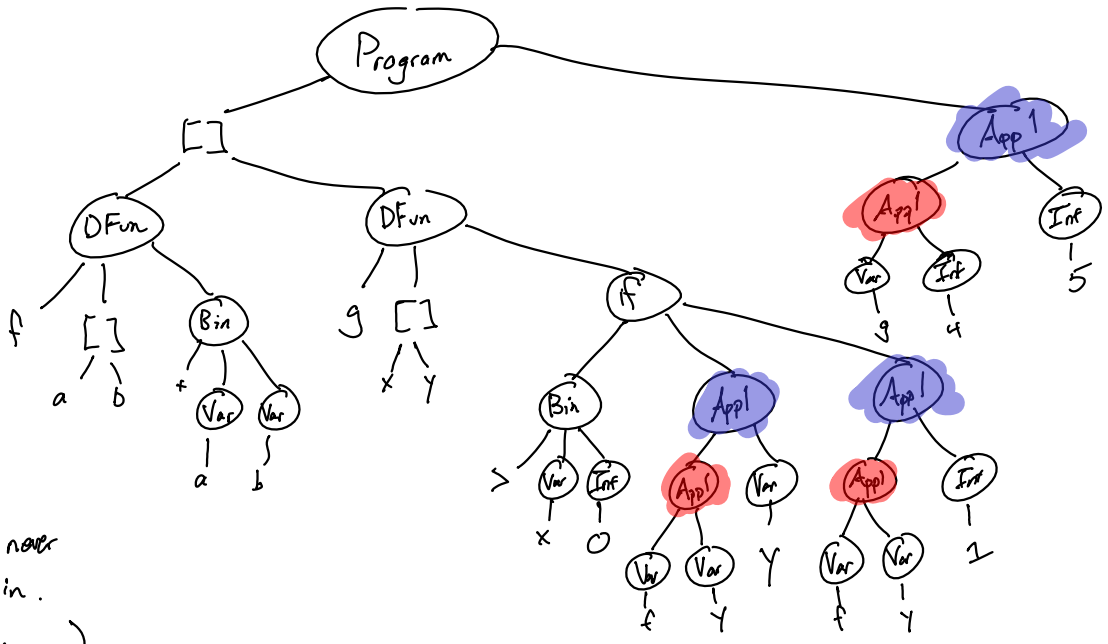


```
def f a b =
  a+b
end
def g x y =
  if x>0 then f y y else f y 1
end
g 4 5
```

Not a tail call

Tail call

Where are the tail calls?



In Floppe, we will never TCO in bird_main. (Only for convenience.)

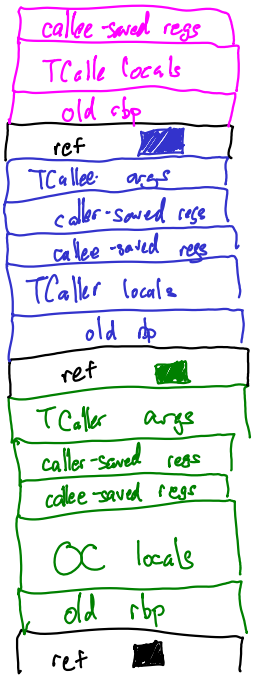
It makes no sense to TCO a closure construction.

Tail Callee

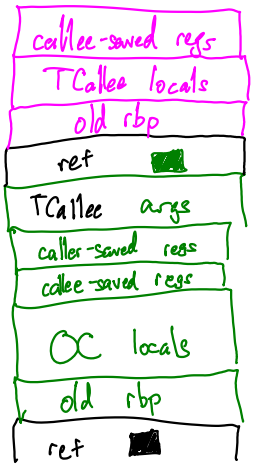
Tail Caller

Original Caller

before calling tail callee



without TCO



with TCO

if # TCaller params < # TCallee params → in Hoopoe: don't TCO

How do we get here?

Assume: # TCaller params ≥ # TCallee params

- Store args for TCallee over the args for TCaller
- Tear down our stack frame (pop rbp)
- jmp to TCallee