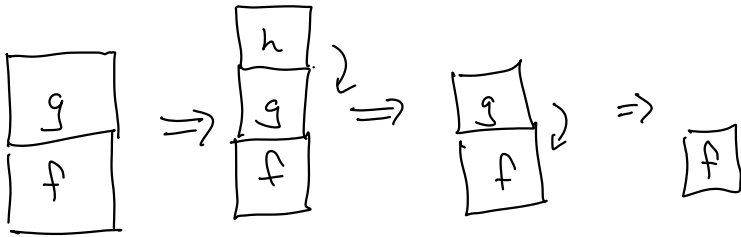


# TCO

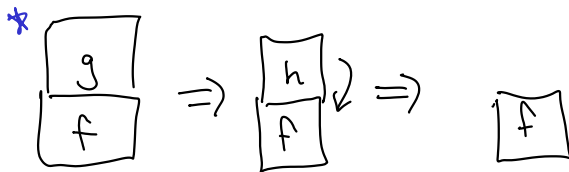
Tail call: call at end of fn

TCO: get rid of caller's frame before making call

w/o TCO



w/ TCO



```
let rec summate n =  
  if n < 1 then 0 else  
    n + summate (n-1)
```

;;

```
let summate n =
```

```
  let rec loop i a =  
    if i ≥ n then a else  
      loop (i+1) (a+i)
```

in

```
  loop 0 0
```

;;

Tail Callee  
 Tail Caller  
 Original Caller

Call tail callee  
 w/o TCO



About to call  
 tail callee

$$\#TCalleeParams = \#TCallerParams$$

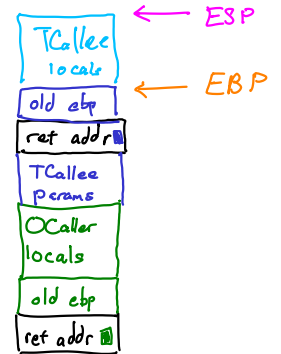


Call tail callee  
 with TCO

1. Removed TCaller locals
2. Replacing TCaller params w/ TCallee params
3. Pop old EBP value into register
4. Jump into TCallee



EBP  
 old ebp



After TCallee  
 jump  


✓



Call tail callee  
w/o TCO

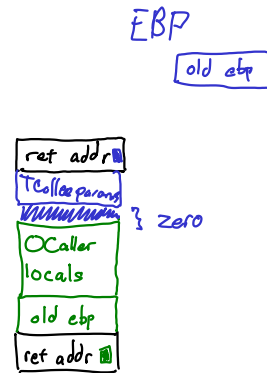


About to call  
tail callee

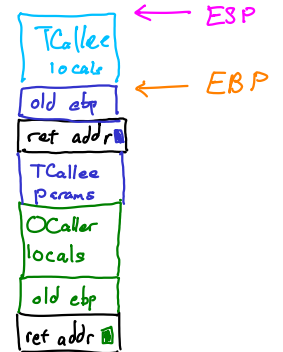


$$\#TCalleeParams \leq \#TCallerParams$$

Call tail callee  
with TCO



After TCallee  
jump



1. Removed TCaller locals
2. Replacing TCaller params w/ TCallee params and zero out other (now unused) memory
3. Pop old EBP value into register
4. Jump into TCallee

Do I tail call?

- Is tail pos? (static)
- TCallee args ≤ TCaller args (dynamic)

\* Argument calling conventions:  
• ECX contains param count

