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

$\text{sum } 5 \Rightarrow \underline{5+}\ \text{sum } (5-1)$

$\Rightarrow 5 + \underbrace{\text{sum } 4}$

$\Rightarrow \boxed{5+} \left(4 \underline{+} \overbrace{\text{sum } (4-1)}\right)$ Work getting done

$\Rightarrow 5 + (4 + \text{sum } 3)$

Work to be done

$\Rightarrow \overbrace{5 + (4 + (3 +}\ \overbrace{\text{sum } (3-1)))}$

$\Rightarrow \dots$

$\Rightarrow 5 + (4 + (3 + (2 + (1 + 0))))$

$\Rightarrow 15$

```
let rec helper acc n = if n = 0 then acc else helper (n + acc) (n - 1)
```

```
helper 0 5 ⟹ helper (5+0) (5-1)
           ⟹ helper 5 4
           ⟹ helper (4+5) (4-1)
           ⟹ helper 9 3
           ⟹ ... ⟹ helper 12 2
           ⟹ ... ⟹ helper 14 1
           ⟹ ... ⟹ helper 15 0
```

Work that needs to be done "after the call"

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

```
let goodSum n =
  let rec helper acc n = if n = 0 then acc else helper (n + acc) (n - 1) in
  helper 0 n
```

No work needs to be done after the call.

```
let rec badRev l =
  match l with
  | [] -> []
  | hd :: tl -> (badRev tl) @ [hd]
```

"Continuation" ⇒ Work to be done "afterwards".

TAIL CALL RECURSION

```
let goodRev l =
  let rec helper acc l =
    match l with
    | [] -> acc
    | hd :: tl -> helper (hd :: acc) tl in
  helper [] l
```

Empty continuation!

Nothing to be done afterwards!