② Evaluating Let expressions

let v = e_1 in e_2

v: variable, e_1: binding

let x = 3+8 in x+9

① Evaluate 3+8

② "store" the value 11 in x

③ Evaluate this: 20

---

let x = 3+8 in x + (let x = 5 in x+2) ⟹ "steps to"

let x = 3+8 in x+7 ⟹

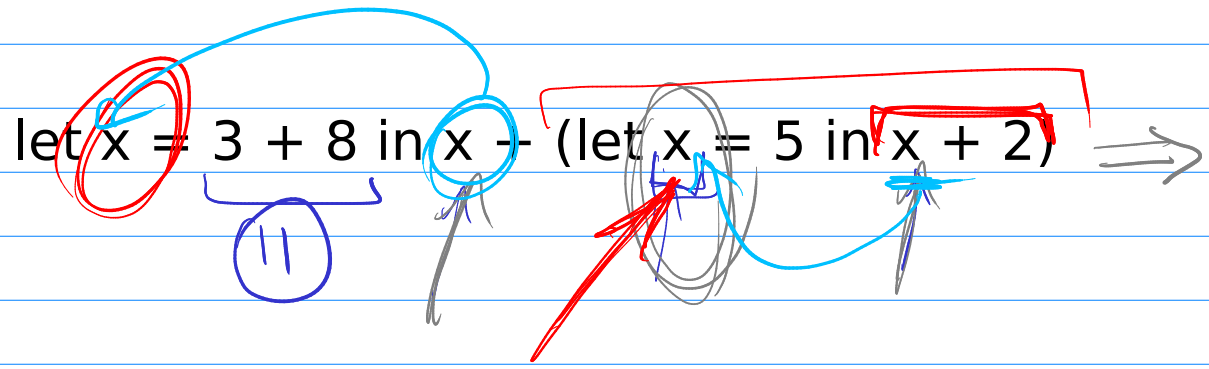
11+7 ⟹ 18

Static scoping / Lexical scoping

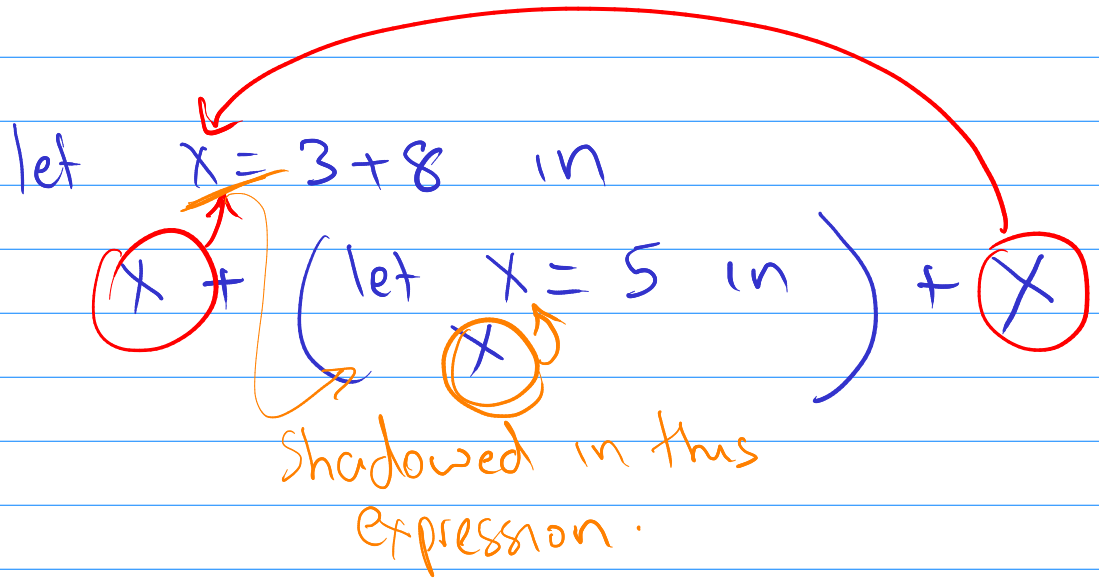
(cf. dynamic scoping)

Rule #3: To evaluate let $x = e_1$ in e_2

1. First evaluate e_1. Say it produces value v.
2. Wherever you see x in e_2, replace it with v. (unshadowed)
3. Finish evaluating e_2.



$11 + (\text{let } x = 5 \text{ in } 11 + 2)$



---

let x = 5+3 in

let x = 9+6 in

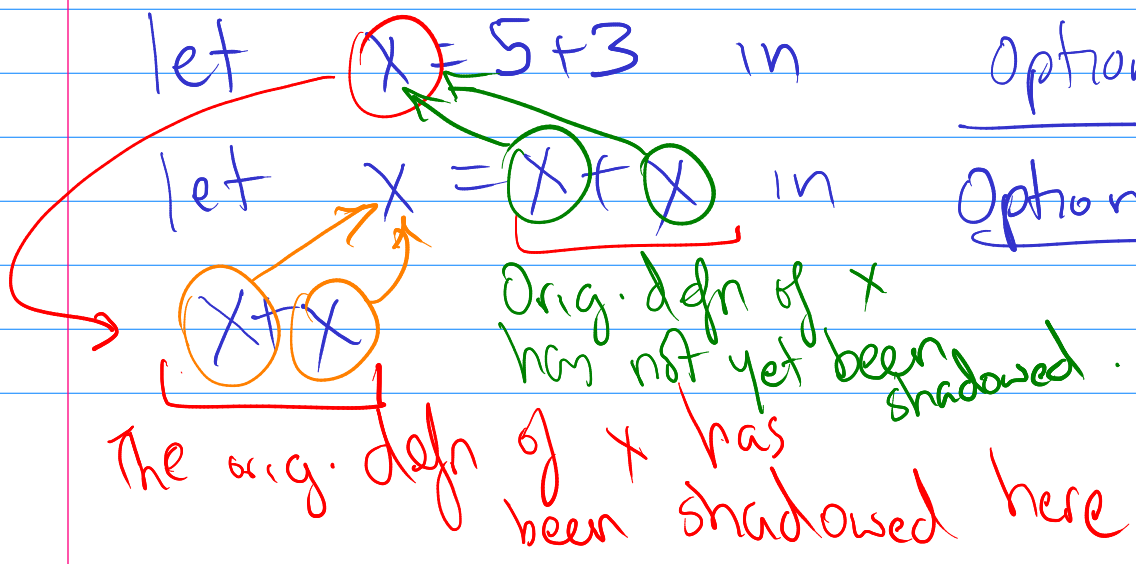
x+x

Option ①:

(5+3)×2 = 16

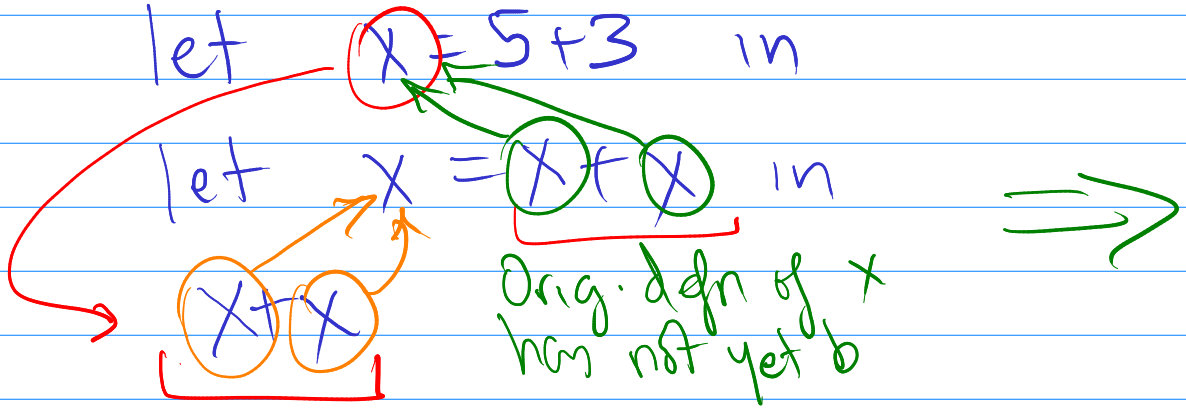
Option ②

(9+6)×2 = 30



Option ①: 32

Option ②: Unbound value x.



let x = 8 in
let x = x+x in ⇒
x+x

let x = 8+8 in ⇒*
x+x

16+16 ⇒ 32

```
let rec consLots n = if n = 0 then [] else n :: consLots (n - 1)
```