

② Evaluating Let expressions

let $v = e_1$ in e_2

variable binding

let $x = 3+8$ in $x+9$

① Evaluate

③ Evaluate this

20

② Store the value in x

let $x = 3+8$ in

"steps to"

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

let $x = 3+8$ in $x+7 \Rightarrow$

$11+7 \Rightarrow 18$

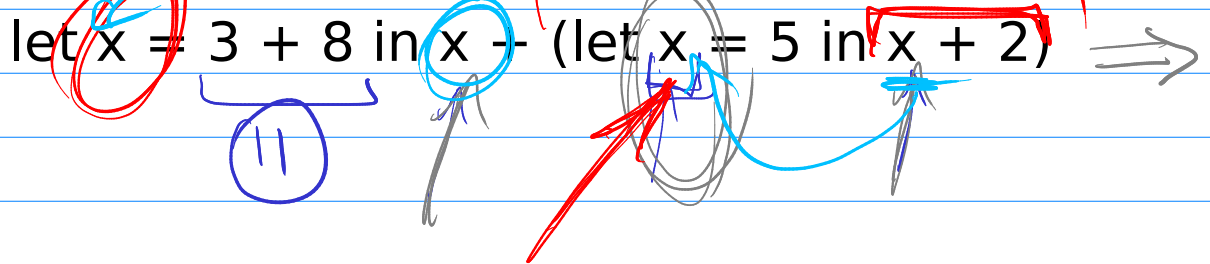
Static scoping / Lexical scoping

(cf. dynamic scoping)

Rule #3: To evaluate $\text{let } x = e_1 \text{ in } e_2$

- 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 .

$\text{let } x = 3 + 8 \text{ in } x + (\text{let } x = 5 \text{ in } x + 2)$ \Rightarrow



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

let $x = 3 + 8$ in

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

Shadowed in this expression.

let $x = 5 + 3$ in

let $x = 9 + 6$ in

$x + x$

Option ①:

$$(5 + 3) \times 2 = 16$$

Option ②

$$(9 + 6) \times 2 = 30$$

let $x = 5 + 3$ in

let $x = x + x$ in

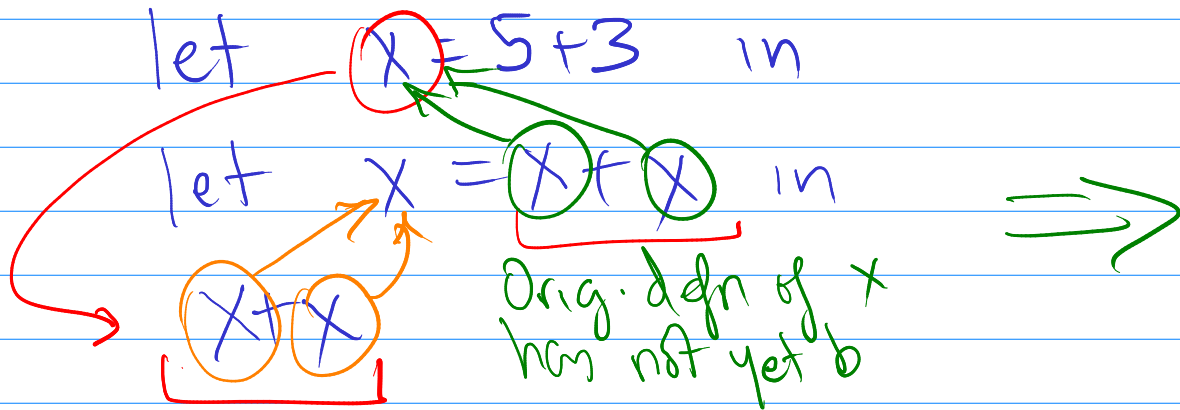
$x + x$

Orig. defn of x has not yet been shadowed.

The orig. defn of x has been shadowed here

Option ①: 32

Option ②: Unbound value x .



let $x = 8$ in

let $x = x + x$ in \Rightarrow

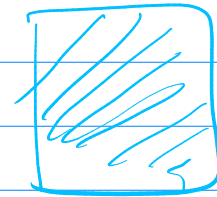
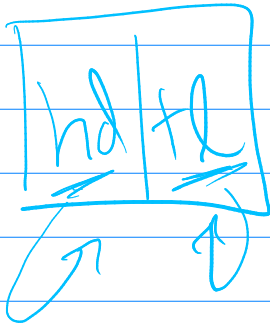
$x + x$

let $x = 8 + 8$ in

$x + x$

$16 + 16 \Rightarrow 32$

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



hd & tl

