Syntax

$$e ::= \bar{v} \mid x \mid (\mathrm{op}\ e \ldots) \mid \mathrm{ifte}\ e\ e\ e \mid (\mathrm{let}\ ((x\ e) \ldots)\ e)$$

Semantic value domain

$$v ::= n \mid b \qquad \text{Val}$$

$$b ::= \mathrm{true} \mid \mathrm{false} \qquad \text{Bool}$$

$n \in \mathbb{N}$ Nat

$x \in \mathrm{Sym}$ Sym

Semantics: $\Gamma \vdash e \Rightarrow v$

$\Gamma : \mathrm{Sym} \xrightarrow{\mathrm{fin}} \mathrm{Val}$

$\Gamma(x)$ is undefined if $x \notin \mathrm{dom}(\Gamma)$

Operations on environments:

$$\Gamma_1, \Gamma_2 \stackrel{\text{df}}{=} \lambda(x)$$
$$\text{if } x \in \text{dom}(\Gamma_1) \text{ then } \Gamma_1(x)$$
$$\text{else } \Gamma_2(x)$$

$$\text{dom}(\Gamma_1, \Gamma_2) \stackrel{\text{def}}{=} \text{dom}(\Gamma_1) \cup \text{dom}(\Gamma_2)$$

Examples

$$\Gamma_1 = \{x \mapsto 5,\ y \mapsto 3,\ z \mapsto 4\}$$
$$\Gamma_2 = \{a \mapsto 6,\ x \mapsto 4,\ y \mapsto 1\}$$

$$\text{dom}(\Gamma_1, \Gamma_2) = \{x, y, z, a\}$$

$$\Gamma_1, \Gamma_2(x) = 5$$
$$\Gamma_1, \Gamma_2(a) = 6$$

Evaluation:

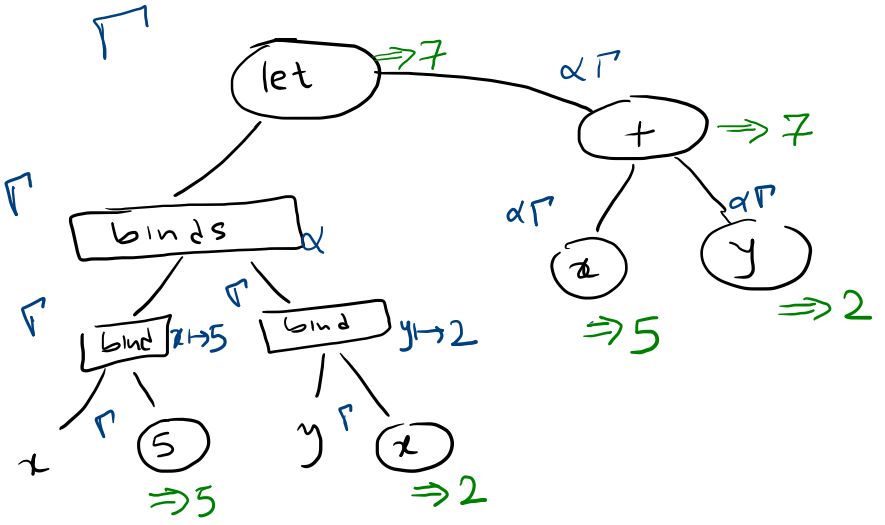
$$\Gamma \vdash e \Rightarrow v$$

$$\frac{}{\Gamma \vdash \bar{v} \Rightarrow v}\ \text{VAL}$$

$$\frac{}{\Gamma, \{x \rightarrow \bar{v}\} \vdash x \Rightarrow v}\ \text{VAR}\quad x \in \text{dom}(\Gamma)$$

$$\frac{\Gamma \vdash e_1 \Rightarrow \bar{v}_1 \qquad \Gamma \vdash e_2 \Rightarrow \bar{v}_2}{\Gamma \vdash (\text{op}\ e_1\ e_2) \Rightarrow v_3}\ \text{OP}\quad \text{op}(v_1, v_2) = v_3$$

$$\frac{\Gamma \vdash \overrightarrow{e_i \Rightarrow v_i} \qquad \overrightarrow{x_i \mapsto v_i}\, \Gamma \vdash e \Rightarrow v}{\Gamma \vdash \text{let } \overrightarrow{x_i = e_i} \text{ in } e \Rightarrow v}\ \text{LET}$$



$e = (\text{let } ([x\ 5]\ [y\ x])\ (+\ x\ y))$

$fvars(e) = x$

$\Gamma = \{x \mapsto 2, y \mapsto 3\}$

$\alpha = \{x \mapsto 5, y \mapsto 2\}$

$\alpha\Gamma = \{x \mapsto 5, y \mapsto 2\}$

Examples :

Lang	Lexical Scope	
C	✓	
Scheme	✓	
Emacs Lisp	X	
TeX	X	
Shell	X	only global vars

Implementation of Envs as procedures:

$Env = Sym \longrightarrow VAL + ERROR$

```
(define empty-env
  (λ (x)
    (error 'empty-env "unbound id ~s" x)))

(define extended-env
  (λ (ids vals env)
    (λ (x)
      (let ([p (locate x ids)])
        (if (= p -1)
            (env x)
            (list-ref vals p))))))
```

```
(define-datatype  env  env?
   [empty-env ()]
   [extended-env (ids (list-of symbol?))
                 (vals (list-of vals!))
                 (env env?)])
```

```
;; apply-env : symbol? ⟶ val?
(define apply-env
```

LEXICAL SCOPE

Syntax

$e ::= \overline{v}$ literals

x identifiers

$+ \; e \; e$ PLUS

$\text{ifte} \; e \; e \; e$ IFTE

$\text{let} \; (x \; e) \ldots \; e$ LET ← LOCAL BINDINGS

$x \in$ Identifiers ID

$v \in$ VALUES

$v ::= n$ Numbers

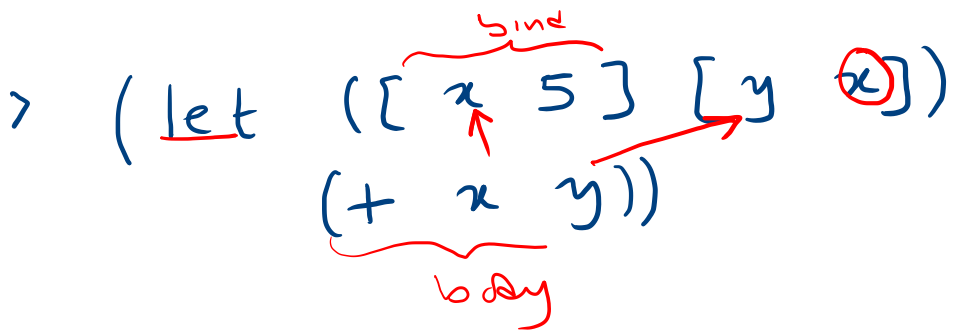
b Boolean values

$b \in$ Boolean $= \{t, f\}$

$n \in$ Numbers

BLOCK STRUCTURED LANGUAGES

```
{   int x = 5
    .
    .
    {
        int x = 0 ;
        .
        .
        x
    }
}
```



Earlier $e \Longrightarrow v$

Now, we need environments

$$\Gamma \vdash e \Longrightarrow v$$

Environment = $ID \xrightarrow{fin} VAL$

Examples

$\Gamma = \{x \mapsto 5,\ y \mapsto 2\}$

$\Gamma(x) = 5$

$dom(\Gamma) = \{x, y\}$

ϕ Empty Env

$\Gamma_1 \; \Gamma_2$ is an environment Γ_3 extension of Γ_2 with Γ_1.

where $dom(\Gamma_3) = dom(\Gamma_1) \cup dom(\Gamma_2)$

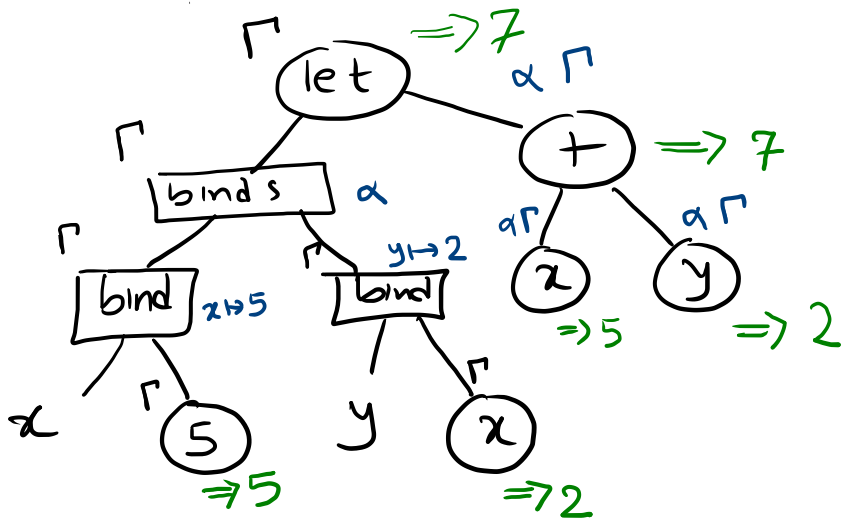
and $\Gamma_3(x) =_{df}$ if $x \in dom(\Gamma_1)$ then $\Gamma_1(x)$
else if $x \in dom(\Gamma_2)$ then $\Gamma_2(x)$
else UNDEFINED

example

$\Gamma_1 = \{x \mapsto 3, y \mapsto 4\}$

$\Gamma_2 = \{x \mapsto 7, z \mapsto 2\}$

$\Gamma_3 \stackrel{df}{=} \Gamma_1 \; \Gamma_2 = \{x \mapsto 3, y \mapsto 4, z \mapsto 2\}$




```
(let ([x 5] [y x])
  (+ x y))
```

([x 5] [y x]) in the context Γ ELABORATES to α.

$\Gamma = \{x \mapsto 2, y \mapsto 3, z \mapsto 4\}$

$\alpha = \{x \mapsto 5, y \mapsto 2\}$

$\alpha\Gamma = \{x \mapsto 5, y \mapsto 2, z \mapsto 4\}$

Evaluation Semantics

Judgements $\Gamma \vdash e \Rightarrow v$

$$\frac{}{\Gamma \vdash \bar{v} \Rightarrow v} \text{ LITERAL}$$

$$\frac{x \in dom(\Gamma) \qquad \Gamma(x) = v}{\Gamma \vdash x \Rightarrow v} \text{ IDENTIFIER}$$

$$\frac{\Gamma \vdash e_1 \Rightarrow v_1 \quad \Gamma \vdash e_2 \Rightarrow v_2}{\Gamma \vdash +\, e_1\ e_2 \Rightarrow v_1 + v_2}\ \text{PLUS}$$

$$\frac{\Gamma \vdash e_1 \Rightarrow \text{true} \quad \Gamma \vdash e_2 \Rightarrow v_2}{\Gamma \vdash \text{ifte}\ e_1\ e_2\ e_3 \Rightarrow v_2}\ \text{IF-TRUE}$$

$$\frac{\Gamma \vdash e_1 \Rightarrow \text{false} \quad \Gamma \vdash e_3 \Rightarrow v_3}{\Gamma \vdash \text{ifte}\ e_1\ e_2\ e_3 \Rightarrow v_3}\ \text{IF-FALSE}$$

$$\frac{(\Gamma \vdash e_1 \Rightarrow v_1) \cdots (\Gamma \vdash e_n \Rightarrow v_m) \quad \alpha = \{x_1 \mapsto v_1 \cdots x_n \mapsto v_m\} \qquad \alpha\,\Gamma \vdash e \Rightarrow v}{\Gamma \vdash \text{let } (x_1\ e_1) \cdots (x_n\ e_n)\ e \Rightarrow v}\ \text{LET}$$

Id	val
x	5
y	2
z	3

API for environments:

1. empty-env : Env?
2. lookup-env : Env? Id? ⟶ Val?
3. extend-env : Env? (listof Id?) (listof Val?) ⟶ Env?