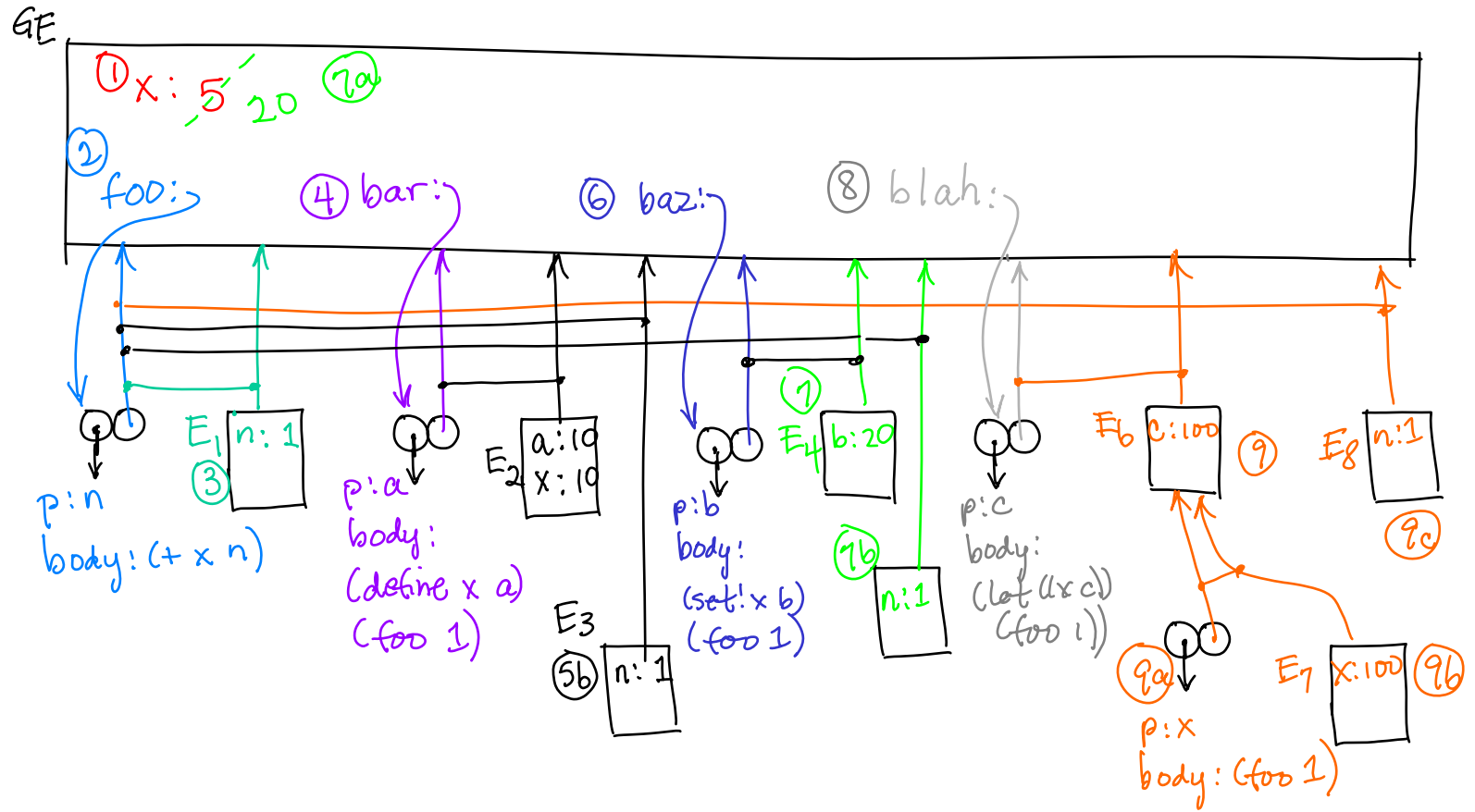


4/6/07 environment diagram example (solution)

- ① (define x 5)
- ② (define (foo n) (+ x n))
- ③ (foo 1) => _____
- ④ (define (bar a) (define x a) (foo 1))
- ⑤ (bar 10) => _____
- ⑥ (define (baz b) (set! x b) (foo 1))
- ⑦ (baz 20) => _____
- ⑧ (define (blah c) (let ((x c)) (foo 1)))
- ⑨ (blah 100)



example (explanation)

① (define x 5)

② (define (foo n)
 (+ x n))

③ (foo 1) => 6

④ (define (bar a)
 (define x a)
 (foo 1))

5. (bar 10) => _____

6. (define (baz b)
 (set! x b)
 (foo 1))

7. (baz 20) => _____

8. (define (blah c)
 (let ((x c))
 (foo 1)))

9. (blah 100) => _____

① add x to GE and set its value to 5

② add foo to GE and set its value to a double linkable whose code ptr points to parameter n and body (+ x n), and whose env ptr points to GE

③ eval (foo 1) | GE — drop a new frame E₁, link E₁'s ptr to foo's defining env (GE) link foo's param n to 1, and eval foo's body in E₁:
 (+ x n) | E₁ look for x in E₁, then GE
 (+ 5 1) | E₁ ⇒ 6

④ add bar to GE and set its value to a double linkable whose code ptr points to parameter a and body (define x a) (foo 1); and whose env ptr points to bar's env (GE)

example (explanation)

1. (define x 5)

2. (define (foo n)
 (+ x n))

3. (foo 1) => 6

4. (define (bar a)
 (define x a)
 (foo 1))

5. (bar 10) => 6

6. (define (baz b)
 (set! x b)
 (foo 1))

7. (baz 20) => _____

8. (define (blah c)
 (let ((x c))
 (foo 1)))

9. (blah 100) => _____

5. eval (bar 10) | $\mathcal{G}E$: drop new frame E_2 , link it to bar's defining env, bind bar's param of a to 10; then eval bar's body in E_2

5a. (define x a) | E_2 adds x to the current env, E_2 , and binds it to the value of a, which is 10

5b. (foo 1) | E_2 drop new frame E_3 , link to foo's env*, bind foo's param to 1, (+ x 1) | E_3 and eval foo's body in E_3
(+ 5 1) | E_3 look for x in E_3 , then $\mathcal{G}E$
| $E_3 \Rightarrow 6$

* Note: Because foo was defined in $\mathcal{G}E$, E_3 points to $\mathcal{G}E$; this behavior is lexical scoping. If E_3 pointed to the environment from which foo was called (E_2), this behavior would be dynamic scoping.

6. add baz to $\mathcal{G}E$ and set its value to a double linkable whose code ptr points to parameter b and body (set! x b) (foo 1); and whose env ptr points to baz's env ($\mathcal{G}E$)

example (explanation)

1. (define x 5)

2. (define (foo n)
 (+ x n))

3. (foo 1) => 6

4. (define (bar a)
 (define x a)
 (foo 1))

5. (bar 10) => 6

6. (define (baz b)
 (set! x b)
 (foo 1))

7. (baz 20) => 21

8. (define (blah c)
 (let ((x c))
 (foo 1)))

9. (blah 100) => _____

① eval (baz 20) | G_E - drop new frame E_4 , link it to baz's defining env (G_E), bind baz's param b to 20; then eval baz's body in E_4

①a (set! x b) | E_4 look for x in E_4 ; it's not there so look in E_4 's enclosing env + find x in G_E ; x in G_E is set to 20.

①b (foo 1) | E_4 drop a new frame E_5 , link to foo's env (G_E), bind foo's param to 1 and eval foo's body in E_5

(+ x 1) | E_5 look for x in E_5 , then G_E
(+ 20 1) | E_5 => 21

⑧ add blah to G_E and set its value to a double bubble with param c , body (let ((x c)) (foo 1)), and env ptr to G_E

example (explanation)

1. (define x 5)
2. (define (foo n) (+ x n))
3. (foo 1) => 6
4. (define (bar a) (define x a) (foo 1))
5. (bar 10) => 6
6. (define (baz b) (set! x b) (foo 1))
7. (baz 20) => 21
8. (define (blah c) (let ((x c)) (foo 1)))
9. (blah 100) => 21

⑨ eval (blah 100) | ΓE - drop a new frame E_b + eval blah's body in E_b
 recall that let desugars into a procedure and an application: (let ((x c)) (foo 1)) \Rightarrow (($\lambda(x)$) (foo 1) c)

⑨a eval ($\lambda(x)$ (foo 1)) | E_b :
 add the double bubble for the anonymous procedure and link it to E_b ; it has a param x and a body (foo 1)

⑨b apply the anonymous procedure to c: drop a new frame E_7 , link the anon proc's x param to c, whose value is found in E_b ; link E_7 to the anon proc's env (E_b)

⑨c eval the anon proc's body in E_7 : (foo 1) | E_7
 drop a new frame E_8 , link to foo's defining env (ΓE), link n to 1, + eval foo's body in E_8
 (+ x n) | E_8 lookup x in E_8 , then ΓE
 (+ 20 1) | $E_8 \Rightarrow 21$