



Liam O'Connor

CSE, UNSW (and data61)

Semester 2 2018

## Misc

- Assignment 1 has been **released**.
- It is due on Tuesday of Week 9, that is **18th September 2018**, 23:55.
- It is worth 30% of your class mark.
- You must complete a base task for 70% of those marks, and **three** of **five** extra tasks, each worth 10%.
- It is possible to score over 100% for this assignment, allowing you to compensate for lost marks elsewhere.
- Assignment tour in the lecture today, maybe overflowing into the extension slot.

## Where we're at

- We refined the abstract M-Machine to a **C-Machine**, with explicit stacks:

$$s \succ e \quad s \prec v$$

- Function application is still executed via substitution:

$$\frac{}{(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \prec v \mapsto_C s \prec e[x := v, f := (\text{Fun } (f.x.e))]}$$

- We're going to extend our C-Machine to replace substitutions with an **environment**, giving us a new **E-Machine**

# Environments

## Definition

An *environment* is a *context* containing equality assumptions about variables.

It can be viewed like a *state* that maps variables to their values, except that a variable's value does not change over time.

$$\frac{}{\bullet \text{ Env}} \quad \frac{\eta \text{ Env}}{x = v, \eta \text{ Env}}$$

We write  $\eta(x)$  to indicate the leftmost value corresponding to  $x$  in  $\eta$ .

Let's change our machine states to include an *environment*:

$$s \mid \eta \succ e \quad s \mid \eta \prec v$$

## First Attempt

First, we'll add a rule for consulting the environment if we encounter a free variable:

$$\frac{}{s \mid \eta \succ x \mapsto_E s \mid \eta \prec \eta(x)}$$

Then, we just need to handle function application.

One broken attempt:

$$\frac{}{(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta) \succ e}$$

We don't know when to remove the variables again!

## First Attempt

First, we'll add a rule for consulting the environment if we encounter a free variable:

$$\frac{}{s \mid \eta \succ x \mapsto_E s \mid \eta \prec \eta(x)}$$

Then, we just need to handle function application.

**One broken attempt:**

$$\frac{}{(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta) \succ e}$$

We don't know when to remove the variables again!

## First Attempt

First, we'll add a rule for consulting the environment if we encounter a free variable:

$$\frac{}{s \mid \eta \succ x \mapsto_E s \mid \eta \prec \eta(x)}$$

Then, we just need to handle function application.

**One broken attempt:**

$$\frac{}{(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta) \succ e}$$

We don't know when to remove the variables again!

## Second Attempt

We will extend our **stacks** to allow us to **save** the old environment to it.

$$\frac{s \text{ Stack} \quad \eta \text{ Env}}{\eta \triangleright s \text{ Stack}}$$

Calling a function, we save the environment to the stack.

---


$$(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E \eta \triangleright s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta) \succ e$$

When the function returns, we restore the old environment, clearing out the new bindings:

$$\frac{}{\eta \triangleright s \mid \eta' \prec v \mapsto_E s \mid \eta \prec v}$$



## Second Attempt

We will extend our **stacks** to allow us to **save** the old environment to it.

$$\frac{s \text{ Stack} \quad \eta \text{ Env}}{\eta \triangleright s \text{ Stack}}$$

Calling a function, we save the environment to the stack.

$$\overline{(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E \eta \triangleright s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta) \succ e}$$

When the function returns, we restore the old environment, clearing out the new bindings:

$$\overline{\eta \triangleright s \mid \eta' \prec v \mapsto_E s \mid \eta \prec v}$$

## Second Attempt

We will extend our **stacks** to allow us to **save** the old environment to it.

$$\frac{s \text{ Stack} \quad \eta \text{ Env}}{\eta \triangleright s \text{ Stack}}$$

Calling a function, we save the environment to the stack.

$$\overline{(\text{Apply } \langle\langle f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E \eta \triangleright s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta) \succ e}$$

When the function returns, we restore the old environment, clearing out the new bindings:

$$\overline{\eta \triangleright s \mid \eta' \prec v \mapsto_E s \mid \eta \prec v}$$

## Simple Example

	○		●		⋈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○		●		⋈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○		●		⋈	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○		●		⋈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○		●		⋈	3
	●▷		●	x = 3, f = ⟨⟨...⟩⟩,	●	⋈ (Plus x (N 1))
(Plus □ (N 1))▷	●▷		●	x = 3, f = ⟨⟨...⟩⟩,	●	⋈ x
(Plus □ (N 1))▷	●▷		●	x = 3, f = ⟨⟨...⟩⟩,	●	⋈ 3
(Plus 3 □)▷	●▷		●	x = 3, f = ⟨⟨...⟩⟩,	●	⋈ (N 1)
(Plus 3 □)▷	●▷		●	x = 3, f = ⟨⟨...⟩⟩,	●	⋈ 1
	●▷		●	x = 3, f = ⟨⟨...⟩⟩,	●	⋈ 4
	○		●		●	⋈ 4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	$\Upsilon$	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	$\Upsilon$	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	$\Upsilon$	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	$\Upsilon$	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	$\Upsilon$	3
	●▷	○	$\Upsilon$	(Plus x (N 1))
(Plus □ (N 1))▷	●▷	○	$\Upsilon$	x
(Plus □ (N 1))▷	●▷	○	$\Upsilon$	3
(Plus 3 □)▷	●▷	○	$\Upsilon$	(N 1)
(Plus 3 □)▷	●▷	○	$\Upsilon$	1
	●▷	○	$\Upsilon$	4
	○	●	$\Upsilon$	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌋	3
●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	x
(Plus □ (N 1))▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	3
(Plus 3 □)▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(N 1)
(Plus 3 □)▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	1
●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	4
	○	●	⌋	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷○	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷○	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷○	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷○	○	●	⌋	3
●▷○	○	●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷●▷○	○	●	⌈	x
(Plus □ (N 1))▷●▷○	○	●	⌋	3
(Plus 3 □)▷●▷○	○	●	⌈	(N 1)
(Plus 3 □)▷●▷○	○	●	⌋	1
●▷○	○	●	⌋	4
○	○	●	⌋	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌋	3
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	x
(Plus □ (N 1))▷●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	3
(Plus 3 □)▷●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(N 1)
(Plus 3 □)▷●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	1
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	4
	○	●	⌋	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌈	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	3
●▷	○	●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷	●▷	○	⌈	x
(Plus □ (N 1))▷	●▷	○	⌈	3
(Plus 3 □)▷	●▷	○	⌈	(N 1)
(Plus 3 □)▷	●▷	○	⌈	1
●▷	○	●	⌈	4
	○	●	⌈	4

Seems to work for **basic examples**, but is there some way to break it?



## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌋	3
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	x
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	3
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(N 1)
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	1
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	4
	○	●	⌋	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌋	3
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	x
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	3
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(N 1)
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	1
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	4
	○	●	⌋	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌞	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3)) ▷	○	●	⌞	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3)) ▷	○	●	⌞	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □) ▷	○	●	⌞	(N 3)
(Ap ⟨⟨...⟩⟩ □) ▷	○	●	⌞	3
	● ▷ ○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	(Plus x (N 1))
(Plus □ (N 1)) ▷	● ▷ ○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	x
(Plus □ (N 1)) ▷	● ▷ ○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	3
(Plus 3 □) ▷	● ▷ ○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	(N 1)
(Plus 3 □) ▷	● ▷ ○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	1
	● ▷ ○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	4
	○	●	⌞	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌞	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌞	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌞	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌞	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌞	3
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	(Plus x (N 1))
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	x
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	3
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	(N 1)
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	1
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	4
	○	●	⌞	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌋	3
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	x
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	3
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(N 1)
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	1
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	4
	○	●	⌋	4

Seems to work for basic examples, but is there some way to break it?

## Simple Example

	○	●	⌞	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌞	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌞	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌞	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌞	3
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	(Plus x (N 1))
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	x
(Plus □ (N 1))▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	3
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	(N 1)
(Plus 3 □)▷	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	1
	●▷○	x = 3, f = ⟨⟨...⟩⟩, ●	⌞	4
	○	●	⌞	4

Seems to work for **basic examples**, but is there some way to break it?

## Simple Example

	○	●	⌈	(Ap (Fun (f.x. (Plus x (N 1)))))
(Ap □ (N 3))▷	○	●	⌈	(Fun (f.x. (Plus x (N 1))))
(Ap □ (N 3))▷	○	●	⌋	⟨⟨f.x. (Plus x (N 1))⟩⟩
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌈	(N 3)
(Ap ⟨⟨...⟩⟩ □)▷	○	●	⌋	3
●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(Plus x (N 1))
(Plus □ (N 1))▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	x
(Plus □ (N 1))▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	3
(Plus 3 □)▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌈	(N 1)
(Plus 3 □)▷	●▷	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	1
●▷	○	x = 3, f = ⟨⟨...⟩⟩, ●	⌋	4
	○	●	⌋	4

Seems to work for **basic examples**, but is there some way to break it?

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3))  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3))  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$   $\llbracket f.x. (Fun (g.y. x)) \rrbracket$

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □)  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □)  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  3

$\mapsto_E$  ●  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ |  $x = 3, f = \llbracket f \dots \rrbracket, \bullet \succ$  (Fun (g.y. x))

$\mapsto_E$  ●  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ |  $x = 3, f = \llbracket f \dots \rrbracket, \bullet \succ$   $\llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$   $\llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □)  $\triangleright$  ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □)  $\triangleright$  ○ | ●  $\succ$  4

$\mapsto_E$  ●  $\triangleright$  ○ |  $y = 4, g = \llbracket g.y. x \rrbracket, \bullet \succ$   $x$

Oh no! We're stuck!



## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll f.x. (Fun (g.y. x)) \gg$

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \succ (Fun (g.y. x))$

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \prec \ll g.y. x \gg$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec \ll g.y. x \gg$

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \ll g.y. x \gg, \bullet \succ x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle f.x. (Fun (g.y. x)) \rangle\rangle$

$\mapsto_E$  (Ap  $\langle\langle f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\langle\langle f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun (g.y. x))

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle g.y. x \rangle\rangle$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle g.y. x \rangle\rangle$

$\mapsto_E$  (Ap  $\langle\langle g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\langle\langle g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, \bullet \succ x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3))  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3))  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\prec$   $\llbracket f.x. (Fun (g.y. x)) \rrbracket$

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □)  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □)  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\prec$  3

$\mapsto_E$  ●  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ |  $x = 3, f = \llbracket f \dots \rrbracket, \bullet \succ$  (Fun (g.y. x))

$\mapsto_E$  ●  $\triangleright$  (Ap □ (N 4))  $\triangleright$  ○ |  $x = 3, f = \llbracket f \dots \rrbracket, \bullet \prec$   $\llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap □ (N 4))  $\triangleright$  ○ | ●  $\prec$   $\llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □)  $\triangleright$  ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □)  $\triangleright$  ○ | ●  $\prec$  4

$\mapsto_E$  ●  $\triangleright$  ○ |  $y = 4, g = \llbracket g.y. x \rrbracket, \bullet \succ$   $x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll f.x. (Fun (g.y. x)) \gg$

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \succ (Fun (g.y. x))$

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \prec \ll g.y. x \gg$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec \ll g.y. x \gg$

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \ll g.y. x \gg, \bullet \succ x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\llbracket f.x. (Fun (g.y. x)) \rrbracket$

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \llbracket f \dots \rrbracket, \bullet \succ (Fun (g.y. x))$

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \llbracket f \dots \rrbracket, \bullet \prec \llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec \llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \llbracket g.y. x \rrbracket, \bullet \succ x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\llbracket f.x. (Fun (g.y. x)) \rrbracket$

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\llbracket f \dots \rrbracket$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \llbracket f \dots \rrbracket$ , ●  $\succ$  (Fun (g.y. x))

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \llbracket f \dots \rrbracket$ , ●  $\prec$   $\llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\llbracket g.y. x \rrbracket$

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\llbracket g.y. x \rrbracket$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \llbracket g.y. x \rrbracket$ , ●  $\succ$   $x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll f.x. (Fun (g.y. x)) \gg$

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \succ (Fun (g.y. x))$

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \prec \ll g.y. x \gg$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec \ll g.y. x \gg$

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \ll g.y. x \gg, \bullet \succ x$

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y. x$ )))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y. x$ )))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y. x$ ))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll f.x. (Fun (g.y. x)) \gg$

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \succ (Fun (g.y. x))$

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \prec \ll g.y. x \gg$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec \ll g.y. x \gg$

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \ll g.y. x \gg, \bullet \succ x$

Oh no! We're stuck!



## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun (f.x. (Fun (g.y. x)))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun (f.x. (Fun (g.y. x))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll f.x. (Fun (g.y. x)) \gg$

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg$ , ●  $\succ$  (Fun (g.y. x))

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg$ , ●  $\prec$   $\ll g.y. x \gg$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll g.y. x \gg$

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \ll g.y. x \gg$ , ●  $\succ$  x

Oh no! We're stuck!

## Closure Capture

○ | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y. x$ )))) (N 3)) (N 4))

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y. x$ )))) (N 3))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y. x$ ))))

$\mapsto_E$  (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\ll f.x. (Fun (g.y. x)) \gg$

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

$\mapsto_E$  (Ap  $\ll f \dots \gg$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \succ (Fun (g.y. x))$

$\mapsto_E$  ● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \ll f \dots \gg, \bullet \prec \ll g.y. x \gg$

$\mapsto_E$  (Ap □ (N 4)) ▷ ○ | ●  $\prec \ll g.y. x \gg$

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\succ$  (N 4)

$\mapsto_E$  (Ap  $\ll g.y. x \gg$  □) ▷ ○ | ●  $\prec$  4

$\mapsto_E$  ● ▷ ○ |  $y = 4, g = \ll g.y. x \gg, \bullet \succ x$

Oh no! We're stuck!

## Closure Capture

$\circ \mid \bullet \succ (\text{Ap } (\text{Ap } (\text{Fun } (f.x. (\text{Fun } (g.y. x)))) (\text{N } 3)) (\text{N } 4))$   
 $\mapsto_E (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid \bullet \succ (\text{Ap } (\text{Fun } (f.x. (\text{Fun } (g.y. x)))) (\text{N } 3))$   
 $\mapsto_E (\text{Ap } \square (\text{N } 3)) \triangleright (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid \bullet \succ (\text{Fun } (f.x. (\text{Fun } (g.y. x))))$   
 $\mapsto_E (\text{Ap } \square (\text{N } 3)) \triangleright (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid \bullet \prec \langle\langle f.x. (\text{Fun } (g.y. x)) \rangle\rangle$   
 $\mapsto_E (\text{Ap } \langle\langle f \dots \rangle\rangle \square) \triangleright (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid \bullet \succ (\text{N } 3)$   
 $\mapsto_E (\text{Ap } \langle\langle f \dots \rangle\rangle \square) \triangleright (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid \bullet \prec 3$   
 $\mapsto_E \bullet \triangleright (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ (\text{Fun } (g.y. x))$   
 $\mapsto_E \bullet \triangleright (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec \langle\langle g.y. x \rangle\rangle$   
 $\mapsto_E (\text{Ap } \square (\text{N } 4)) \triangleright \circ \mid \bullet \prec \langle\langle g.y. x \rangle\rangle$   
 $\mapsto_E (\text{Ap } \langle\langle g.y. x \rangle\rangle \square) \triangleright \circ \mid \bullet \succ (\text{N } 4)$   
 $\mapsto_E (\text{Ap } \langle\langle g.y. x \rangle\rangle \square) \triangleright \circ \mid \bullet \prec 4$   
 $\mapsto_E \bullet \triangleright \circ \mid y = 4, g = \langle\langle g.y. x \rangle\rangle, \bullet \succ x$

Oh no! We're stuck!

## Something went wrong!

Returning functions as a result means that the function's body expression escapes the scope of bound variables that existed where it is defined:

**(let  $x = 3$  in **refun**  $f$   $y = x + y$ ) 5**

The function value  $\langle\langle f.y. x + y \rangle\rangle$ , when it is applied, does not “remember” that  $x = 3$  when the function was defined.

**Solution:** Store the environment inside the function value!

$$\frac{}{s \mid \eta \succ (\text{Recfun } (f.x. e)) \mapsto_E s \mid \eta \prec \langle\langle \eta, f.x. e \rangle\rangle}$$

This type of function value is called a *closure*.

---

$(\text{Apply } \langle\langle \eta', f.x. e \rangle\rangle \square) \triangleright s \mid \eta \succ v \mapsto_E \eta \triangleright s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta') \succ e$

Store the  
old env. in  
the stack

Retrieve the new  
env. from the closure

$$(\text{Apply } \langle\langle \eta', f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E \eta \triangleright s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta') \succ e$$

Store the  
old env. in  
the stack

Retrieve the new  
env. from the stack

$$(\text{Apply } \langle\langle \eta', f.x. e \rangle\rangle \square) \triangleright s \mid \eta \prec v \mapsto_E \eta \triangleright s \mid (x = v, f = \langle\langle f.x. e \rangle\rangle, \eta') \succ e$$

Store the  
old env. in  
the stack

Retrieve the new  
env. from the closure

## Our Example

○ | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))

(Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))

(Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))

(Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$

(Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)

(Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3

● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y. x$ ))

● ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec \langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$

(Ap □ (N 4)) ▷ ○ | ●  $\prec \langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$

(Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)

(Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4

● ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ x$

● ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec 3$

○ | ●  $\prec$  3



## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ$   $x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec$  3
- | ●  $\prec$  3

## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec \langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec \langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec 3$
- | ●  $\prec$  3

## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ$   $x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec$  3
- | ●  $\prec$  3

## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec 3$
- | ●  $\prec$  3

## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec 3$
- | ●  $\prec$  3

## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec 3$
- | ●  $\prec$  3

## Our Example

- | ●  $\succ$  (Ap (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3)) (N 4))
- (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Ap (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ )))) (N 3))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (Fun ( $f.x.$  (Fun ( $g.y.$   $x$ ))))
- (Ap □ (N 3)) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle \bullet, f.x. (Fun (g.y. x)) \rangle\rangle$
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\succ$  (N 3)
- (Ap  $\langle\langle \bullet, f \dots \rangle\rangle$  □) ▷ (Ap □ (N 4)) ▷ ○ | ●  $\prec$  3
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \succ$  (Fun ( $g.y.$   $x$ ))
- ▷ (Ap □ (N 4)) ▷ ○ |  $x = 3, f = \langle\langle f \dots \rangle\rangle, \bullet \prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap □ (N 4)) ▷ ○ | ●  $\prec$   $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\succ$  (N 4)
- (Ap  $\langle\langle (x = 3, f = \dots, \bullet), g.y. x \rangle\rangle$  □) ▷ ○ | ●  $\prec$  4
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \succ x$
- ▷ ○ |  $y = 4, g = \langle\langle g.y. x \rangle\rangle, x = 3, f = \dots, \bullet \prec 3$
- | ●  $\prec$  3

# Refinement

- We already sketched the proof that shows that each C-machine execution has a corresponding M-machine execution (**refinement**).
- This means that any functional correctness (not security or cost) property we prove about all M-machine executions of a program apply just as well to any C-machine executions of the same program.
- Now we want to prove that each E-machine execution has a corresponding C-machine execution (and therefore a M-machine execution).



## Ingredients for Refinement

Once again, we want the same ingredients to prove a simulation proof that we did in the previous refinement. That is, we must define an **abstraction function**  $\mathcal{A}$  that converts E-machine states to C-machine states, such that:

- Each initial state in the E-machine is mapped to an initial state in the C-Machine.
- Each final state in the E-machine is mapped to a final state in the C-Machine.
- For each transition from one state to another in the E-machine, either there exists a corresponding transition in the C-Machine, or the two states map to the same C-machine state.

## How to define $\mathcal{A}$ ?

- Our abstraction function  $\mathcal{A}$  applies the environment  $\eta$  as a **substitution** to the current expression, and to the stack, starting at the left.
- If any environment is encountered in the stack, switch to substituting with that environment instead.
- E-Machine values are converted to C-Machine values merely by applying the environment inside closures as a substitution to the expression inside the closure.

With such a function definition, it is trivial to prove that each E-Machine transition has a corresponding transition in the C-Machine, as it is 1:1.

## How to define $\mathcal{A}$ ?

- Our abstraction function  $\mathcal{A}$  applies the environment  $\eta$  as a **substitution** to the current expression, and to the stack, starting at the left.
- If any environment is encountered in the stack, switch to substituting with that environment instead.
- E-Machine values are converted to C-Machine values merely by applying the environment inside closures as a substitution to the expression inside the closure.

With such a function definition, it is trivial to prove that each E-Machine transition has a corresponding transition in the C-Machine, as it is 1:1.

## How to define $\mathcal{A}$ ?

- Our abstraction function  $\mathcal{A}$  applies the environment  $\eta$  as a **substitution** to the current expression, and to the stack, starting at the left.
- If any environment is encountered in the stack, switch to substituting with that environment instead.
- E-Machine values are converted to C-Machine values merely by applying the environment inside closures as a substitution to the expression inside the closure.

With such a function definition, it is trivial to prove that each E-Machine transition has a corresponding transition in the C-Machine, as it is 1:1.

## How to define $\mathcal{A}$ ?

- Our abstraction function  $\mathcal{A}$  applies the environment  $\eta$  as a **substitution** to the current expression, and to the stack, starting at the left.
- If any environment is encountered in the stack, switch to substituting with that environment instead.
- E-Machine values are converted to C-Machine values merely by applying the environment inside closures as a substitution to the expression inside the closure.

With such a function definition, it is trivial to prove that each E-Machine transition has a corresponding transition in the C-Machine, as it is 1:1.

## How to define $\mathcal{A}$ ?

- Our abstraction function  $\mathcal{A}$  applies the environment  $\eta$  as a **substitution** to the current expression, and to the stack, starting at the left.
- If any environment is encountered in the stack, switch to substituting with that environment instead.
- E-Machine values are converted to C-Machine values merely by applying the environment inside closures as a substitution to the expression inside the closure.

With such a function definition, it is trivial to prove that each E-Machine transition has a corresponding transition in the C-Machine, as it is 1:1.