Structure and Interpretation of Computer Programs

COMP200
TODAY
Remember

• Mutation
  • \texttt{set!} to change the value associated with a variable
  • \texttt{set-car!} and \texttt{set-cdr!} to change the values of parts of a list

• The notion of time and context

• From functional programming to state-based programming

• \textbf{Danger}: more bugs and errors!

• Broke the substitution model, any replacements?
TODAY

How does this work?

(define make-counter
  (lambda (n)
    (lambda () (set! n (+ n 1)) n)))

(define ca (make-counter 0))
(ca) ;==> 1
(ca) ;==> 2
(define cb (make-counter 0))
(cb) ;==> 1
(ca) ;==> 3

; ca and cb are independent
TODAY

Environment Model (EM)

- A precise, completely mechanical description of:
  - name-rule looking up the value of a variable
  - define-rule creating a new definition of a variable
  - set!-rule changing the value of a variable
  - lambda-rule creating a procedure
  - application applying a procedure

- Enables analyzing arbitrary scheme code.
  example: make-counter

- Basis for implementing a scheme interpreter
  - for now: draw EM state with boxes and pointers
  - later on: implement with code
ENVIRONMENT MODEL
A Shift in Our Viewpoint

As we introduce the environment model, we are going to shift our viewpoint on computation:

• **Variable:**
  • **OLD** – name for value
  • **NEW** – place into which one can store things

• **Procedure:**
  • **OLD** – functional description
  • **NEW** – object with inherited context

• **Expressions:**
  • Now **only** have meaning with respect to an environment
FRAME
A Table of Bindings

- **binding**: a pairing of a name and a value

  **Example**: $x$ is bound to 15 in frame A
  $y$ is bound to (1 2) in frame A
  the value of the variable $x$ in frame A is 15
Environment E1 consists of frames A and B.

Environment E2 consists of frame B only.
  - A frame may be shared by multiple environments.
**ENVIRONMENT**

*A Sequence of Frames*

- Environment E1 consists of frames A and B
- Environment E2 consists of frame B only
  - A frame may be shared by multiple environments

---

![Diagram](image-url)
ENVIRONMENT MODEL

Evaluation

• All evaluation occurs in an environment.
  • The current environment changes when the interpreter applies a procedure.

• The top environment is called the global environment (GE)
  • Only the GE has no enclosing environment.

• To evaluate a combination:
  • Evaluate the sub-expressions in the current environment
  • Apply the value of the first to the values of the rest
**ENVIRONMENT MODEL**

**Name Rule**

- A name $x$ evaluated in environment $E$ gives the value of $x$ in the first frame of $E$ where $x$ is bound.
- In $E_1$, the binding of $x$ in frame $A$ shadows the binding of $x$ in $B$.

$$\begin{align*}
  z \mid_{GE} &\Rightarrow 10 &
  x \mid_{GE} &\Rightarrow 3 \\
  z \mid_{E1} &\Rightarrow 10 &
  x \mid_{E1} &\Rightarrow 15
\end{align*}$$
ENVIRONMENT MODEL

**define Rule**

- A `define` special form evaluated in environment \( E \) creates or replaces a binding in the first frame of \( E \)

\[
\text{(define } z \ 20) \mid_{GE}
\]
ENVIRONMENT MODEL

define Rule

• A define special form evaluated in environment E creates or replaces a binding in the first frame of E

\[
(\text{define } z \ 20) |_{GE}
\]
A `define` special form evaluated in environment `E` creates or replaces a binding in the first frame of `E`.


define Rule

Let's consider the following expressions evaluated in different environments:

- `(define z 20) |_{GE}`
- `(define z 25) |_{E1}`

The diagram illustrates the binding process in different environments:

- In `GE`, after `(define z 20)`, `z` is bound to 20.
- In `E1`, after `(define z 25)`, `z` is bound to 25.
ENVIRONMENT MODEL

define Rule

- A define special form evaluated in environment E
  creates or replaces a binding in the first frame of E

(define z 20) \mid_{GE}

(define z 25) \mid_{E1}

z \mid_{GE} => 20

z \mid_{E1} => 25
ENVIRONMENT MODEL

**set! Rule**

- A `set!` of variable `x` evaluated in environment `E` changes the binding of `x` in the first frame of `E` where `x` is bound.

\[(\text{set! } z \ 20) \mid_{GE}\]
ENVIRONMENT MODEL

set! Rule

• A set! of variable x evaluated in environment E changes the binding of x in the first frame of E where x is bound

(set! z 20) \mid_{GE}
ENVIRONMENT MODEL

set! Rule

- A set! of variable x evaluated in environment E changes the binding of x in the first frame of E where x is bound.

\[(\text{set! } z \ 20) \ |_{GE}\]
\[(\text{set! } z \ 25) \ |_{E1}\]

Diagram:

- A
  - x: 15
  - y:
  - GE
    - B
      - z: 10
      - 20
      - x: 3
    - E1
      - 1
      - 2
**ENVIRONMENT MODEL**

**set! Rule**

- A *set!* of variable $x$ evaluated in environment $E$ changes the binding of $x$ in the first frame of $E$ where $x$ is bound.

$$ \text{(set! } z \ 20) \ |_{GE} $$  

$$ \text{(set! } z \ 25) \ |_{E_1} $$