6.001 Recitation 12: Mutation!

21 March 2007
RI: Gerald Dalley (dalleyg@mit.edu)
http://people.csail.mit.edu/dalleyg/6.001/SP2007/

Modifying Bindings

(set! <name> <value>) \rightarrow undefined
– Looks for the binding of <name> and changes the binding to the value of the <value> expression
Example:
(define x 10)
x \rightarrow 10
(set! x (* 10 20))
x \rightarrow \_200
– Is set! a special form?

Modifying Pairs

(set-car! <pair> <val>) \rightarrow undefined
(set-cdr! <pair> <val>) \rightarrow undefined
– Change the car/cdr part of the cons cell <pair> to <val>. Example:
(define x (list 1 2))
x \rightarrow (1 2)
(set-car! x 3)
x \rightarrow (3 2)
– Are set-car! and set-cdr! special forms?

Thought Question

• What is the difference between changing a binding (using set!) and changing an object (using set-car! or set-cdr!)?
Warmup & Subtle Points

1. Write a Scheme expression that makes the structure (without using mutation!).
2. Write what Scheme prints for the structure (if you can).

3. Show how the mutation affects the box-and-pointer diagram and the printed representation, assuming the structure is named \textit{x}.
3. Show how the mutation affects the box-and-pointer diagram and the printed representation, assuming the structure is named \texttt{x}.

\begin{verbatim}
(set-car! (cddr x) (caar x))
\end{verbatim}

1. Write a Scheme expression that makes the structure (without using mutation!).
2. Write what Scheme prints for the structure (if you can).

\begin{verbatim}
(define x (let* ((bottom '(x)) (middle (list bottom bottom)) (cons middle (cdr middle))))
\end{verbatim}

; \texttt{x} \Rightarrow (((\texttt{x}) (\texttt{x})) (\texttt{x}))

3. Show how the mutation affects the box-and-pointer diagram and the printed representation, assuming the structure is named \texttt{x}.

\begin{verbatim}
(set-cdr! (first x) (second x))
\end{verbatim}

1. Write a Scheme expression that makes the structure.
2. Write what Scheme prints for the structure (if you can).

\begin{verbatim}
(define x (let* ((lower (cons '() '())) (upper (cons lower lower)) (set-car! lower upper) (set-cdr! lower upper) upper))
\end{verbatim}

; \texttt{x} \Rightarrow ...messy...
3. Show how the mutation affects the box-and-pointer diagram and the printed representation, assuming the structure is named \texttt{x}.

\begin{verbatim}
(set-car! (cdr x) '())
(set-cdr! (car x) '())
\end{verbatim}

1. Draw a box-and-pointer representation of the expression's value.

\begin{verbatim}
(let ((w (list 6 7 8)))
  (set-car! w w)
  (set! w (list w w))
  w))
\end{verbatim}

2. Show how the mutation affects the box-and-pointer diagram and the printed representation, assuming the value of the expression is named \texttt{x}.

\begin{verbatim}
(set-car! (car x) (cddr x))
\end{verbatim}

1. Draw a box-and-pointer representation of the expression's value.

\begin{verbatim}
(let ((y '((a) (b))))
  (set-cdr! (first y) y)
  (set-car! (second y) (cdr y))
  (set! y (car y))
  y))
\end{verbatim}

2. Show how the mutation affects the box-and-pointer diagram and the printed representation, assuming the value of the expression is named \texttt{x}.

\begin{verbatim}
(set-cdr! x (third x))
(set-cdr! (cdr x) nil)
\end{verbatim}

What does mystery do?

\begin{verbatim}
(define (mystery x)
  (define (loop x y)
    (if (null? x)
        y
        (let ((temp (cdr x)))
          (set-cdr! x y)
          (loop temp x)))
    y))

(define a (list 1 2 3 4))
(a ==> (1 2 3 4))
(define b (mystery a))
(a ==> (1))
b ==> (4 3 2 1)
\end{verbatim}