

```

;;; Problem 2.4
(define (my-cdr z) (z (lambda (p q) q)))

```

```

;;; Problem 2.23
(define (my-for-each x y)
  (let ((f (x (car y))))
    (if (null? (cdr y))
        true
        (my-for-each x (cdr y)))))

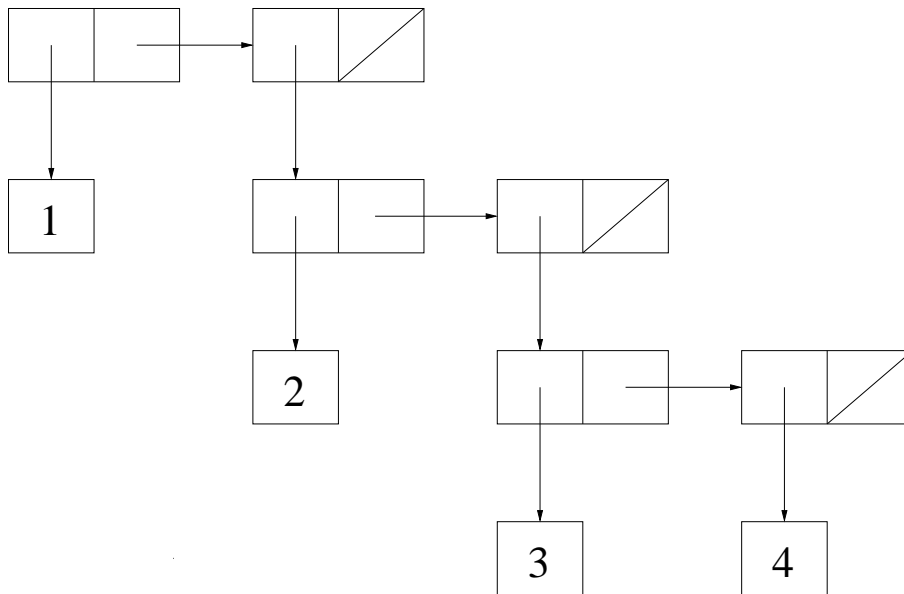
```

```

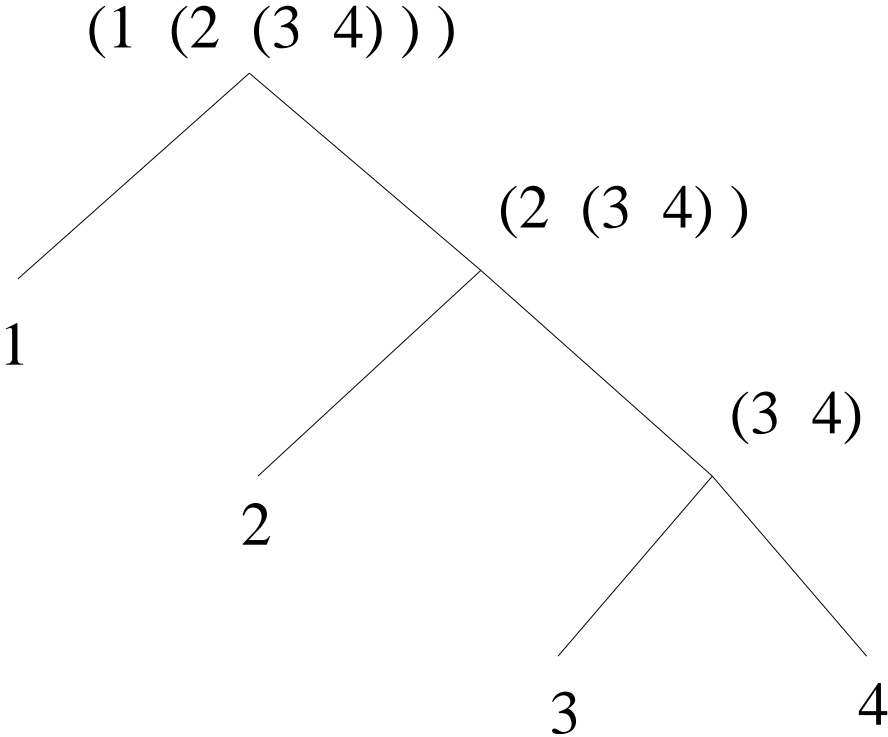
;;; Problem 2.24
Interpreter: (1 (2 (3 4)))

```

Box-and-Pointer Diagram:



Tree:



```

;;; Problem 2.28
(define (fringe l)
  (cond ((null? l) '())
        ((pair? (car l)) (append (fringe (car l)) (fringe (cdr l))))
        (else (cons (car l) (fringe (cdr l))))))

```

```

;;; Problem 2.33
(define (my-map p sequence)
  (accumulate (lambda (x y) (cons (p x) y)) nil sequence))

```

```

(define (my-append seq1 seq2)
  (accumulate cons seq2 seq1))

```

```

(define (my-length sequence)
  (accumulate (lambda (x y) (+ 1 y)) 0 sequence))

```

```

;;; Problem 2.54
(define (equal? a b)
  (cond ((and (null? a) (null? b)) true)
        ((null? a) false)
        ((null? b) false)
        ((and (symbol? a) (symbol? b)) (eq? a b))
        ((symbol? a) false)
        ((symbol? b) false)
        (else (and (equal? (car a) (car b))
                    (equal? (cdr a) (cdr b))))))

```

```

;;; Problem 2.55
'abracadabra evaluates to (quote abracadabra), a function call. When quote
is applied to this, (quote abracadabra) becomes a data object. The sym-
bols 'quote and 'abracadabra are placed in a list, and the car of that list
is 'quote.

```

```

;;; Problem 2.56
(define (exponentiation? x) (eq? (car x) '**))

(define (base x) (cadr x))

(define (exponent x) (caddr x))

(define (make-exponentiation x y) (list '** x y))

;;; Within differentiation program:
((exponentiation? exp)
 (make-product (exponent exp)
               (make-product (make-exponentiation (base exp) (- (exponent exp)
1))
                             (deriv (base exp) var))))

;;; Pre-project exercise
(define (flatten2 l)
  (define (flatten2-helper length l)
    (if (= length 0) nil
        (append (car l) (flatten2-helper (- length 1) (cdr l)))))
  (flatten2-helper (length l) l))

```