

```

;;; The Little Schemer, 4ª ed.
;;; Abrantes Araújo Silva Filho
;;; abrantesasf@gmail.com

;;; Chapter 2: Do It, Do It Again, and Again, and Again...

;; LAT é uma list de atoms:
;; -----
;; lat: (lat? l)
;;      - retorna T se l é uma lista composta somente por atoms
;;      - retorna T se l é uma lista vazia (), pois não contém outras listas
;;      - retorna NIL se l for um atom
;;      - retorna NIL se l conter outra lista
;;
;; Avaliar: (lat? '(pao com manteiga))
;; 1ª CHAMADA: (pao com manteiga)
;; 1) o argumento é atom? ==> NÃO
;; 2) o argumento é ()?   ==> NÃO
;; 3) (car l) é atom?     ==> SIM
;; 4) chamada recursiva com (cdr l)
;; 2ª CHAMADA: (com manteiga)
;; 5) o argumento é atom? ==> NÃO
;; 6) o argumento é ()?   ==> NÃO
;; 7) (car l) é atom?     ==> SIM
;; 8) chamada recursiva com (cdr l)
;; 3ª CHAMADA: (manteiga)
;; 9) o argumento é atom? ==> NÃO
;; 10) o argumento é ()?  ==> NÃO
;; 11) (car l) é atom?    ==> SIM
;; 12) chamada recursiva com (cdr l)
;; 4ª CHAMADA: ()
;; 13) o argumento é atom? ==> NÃO
;; 14) o argumento é ()?   ==> SIM
;; 15) retorna T para a 3ª CHAMADA
;; 3ª CHAMADA:
;; 16) retorna T para a 2ª CHAMADA
;; 2ª CHAMADA:
;; 17) retorna T para a 1ª CHAMADA
;; 1ª CHAMADA:
;; 18) retorna T para o usuário
;;
;; Avaliar: (lat? '(casa (carro)))
;; 1ª CHAMADA: (casa (carro))
;; 1) o argumento é atom? ==> NÃO
;; 2) o argumento é ()?   ==> NÃO
;; 3) (car l) é atom?     ==> SIM
;; 4) chamada recursiva com (cdr l)
;; 2ª CHAMADA: ((carro))
;; 5) o argumento é atom? ==> NÃO
;; 6) o argumento é ()?   ==> NÃO
;; 7) (car l) é atom?     ==> NÃO
;; 8) else retorna NIL para a 1ª CHAMADA
;; 1ª CHAMADA:
;; 9) retorna NIL para o usuário

;; MEMBER verifica se um atom está contido em uma lat:
;; -----
;; member: (member? a lat)
;;      - retorna T se a está contido em lat

```

```
;;          - busca por lista NÃO É CONFIÁVEL devido ao eql?
;;
;; Avaliar: (member? 'casa '(carro casa))
;; 1ª CHAMADA: casa (carro casa)
;;   1) lat é vazia ()          ==> NÃO
;;   2) a é igual à (car lat)? ==> NÃO
;;   3) chamada recursiva com: a (cdr lat)
;; 2ª CHAMADA: casa (casa)
;;   4) lat é vazia ()          ==> NÃO
;;   5) a é igual à (car lat)? ==> SIM
;;   6) retorna T para a 1ª CHAMADA
;; 1ª CHAMADA:
;;   7) retorna T para o usuário

;; O PRIMEIRO MANDAMENTO:
;; -----
;; SEMPRE UTILIZE "null?" como a primeira questão em qualquer função!
```