Patterns of Recursion

- *Linear recursion*: recursing in only one "dimension" or "axis"
  - In Scheme, this typically happens when processing lists at the "top-level" only (i.e., along their length only, generally using *cdr*)
- *Star recursion*/*Tree recursion*: recursing along multiple dimensions or axes
  - In Scheme, this typically happens when processing lists at all levels (i.e., along both their length and their depth, generally using both *cdr* and *car*)

Simple Linear-Recursive Procedure

```
(define name
  (lambda (formal_1 ... formal_n)
    (cond
      (null-case-test
        null-case-consequent)
      (else
        recurse-on-cdr-alternate))))
```

Simple Star-Recursive Procedure

```
(define name
  (lambda ($formal_1$ ... $formal_n$)
    (cond
      (null-case-test
        null-case-consequent)
      (non-atomic-car-case-test
        recurse-on-car-cdr-consequent)
      (else
        recurse-on-cdr-alternate))))
```

Simple Linear-Recursive Procedure

```
(define name
  (lambda (formal_1 ... formal_n)
    (if
      null-case-test
      null-case-consequent
      recurse-on-cdr-alternate)))
```

Simple Star-Recursive Procedure

```
(define name
  (lambda ($formal_1$ ... $formal_n$)
    (if
      null-case-test
      null-case-consequent
      (if
        non-atomic-car-case-test
        recurse-on-car-cdr-consequent
        recurse-on-cdr-alternate))))
```