

(22)

(oddp 3) t 3 is an odd no

(greater 24 27) t arguments are successively larger from left to right

(lessp 5 3 12) nil not smaller from left to right

(null nil) t nil is an empty list

Condinal →
 Predicates are the one-way but Condinal is a two-way i.e. it works just like the if-then-else construct.

Syntax →
 (Cond (<test₁> <action₁>
 <test₂> <action₂>

(<test_k> <action_k>)

i.e. Each (<test_i> <action_i>) where $i=1, \dots, k$ is called a clause. Each clause consists of a test portion and an action or result-portion. Cond is used to define conditions.

for example →

(23)

→ (defun maximum2 (a b) t
 (Cond ((> a b) a)
 (t b)))
 → then
 MAXIMUM2
 →
 → (maximum2 234 320)
 320
 →

Iteration & Recursion →

Iteration → Iteration is used for decision-making. It depends on condition & take a proper action depends on specified-condition.

Syntax →
 (do (<Var₁ Val₁> <Var-update₁>
 <Var₂ Val₂> <Var-update₂>

|
 |
 |
 (<test> <return-value>
 <S-expressions>))

for example →

→ (defun factorial (n)
 (do ((Count n (- Count 1))
 (Product n (* Product (- Count 1))
 ((equal 0 Count) Product)))