Notes lecture 10

1. **EVAL:**

* **EVAL subr 1 (f): e**
* The form is evaluated 2 times:
  + First, it is evaluated as an argument of EVAL
  + Second, it is evaluated by EVAL
* LISP does not evaluate first elem in a form
  + So even though Q = CAR, we cannot write (Q ‘(A B C)) 🡪 *we get an error*

1. **Functional forms:** (function f) = #'f
2. **APPLY & FUNCALL**

* allow the application of a function on a set of parameters that can be synthesized dynamically
* **APPLY lsubr 2 (ff lp): e** = apply a function to parameters provided as a list
  + (APPLY #'MAX '(1 2 3)) 🡪 3
* **FUNCALL lsubr 1- (f ... f): e**
  + allows the application of a function (or expression) resulting from the evaluation of a functional form ff **on a fixed number of parameters**
  + (SETQ Q 'CAR)

(SETQ P 'Q)

(FUNCALL (EVAL P) '(A B C)) = A

1. **LAMBDA expressions**

* Use when:
  + a one-time function is far too simple to be defined
  + the function to be applied must be synthesized dynamically
* **(LAMBDA (p1 p2 … pn)**

**form1**

**form2**

**...**

**form**

**)**

* Usage: **((LAMBDA l f1 f2 ... fm) par1 par2 ... parn)**
* (FUNCALL #'(LAMBDA (L) (CAR L)) '(A B C)) = A

1. **Labels:**

* **(LABELS bindings body)**
  + **bindings** = list containing function definitions (without the DEFUN particle)
  + **body** = program code in which definitions above are effective

1. **Using LAMBDA expressions to avoid repeated calls:**

The whole context that contains the 2 recursive calls is wrapped up in a lambda

function, having as parameter the recursive call.

1. **Generators. Functional arguments**

A white paper with black text

Description automatically generated

1. **MAP functions**

* The role of MAP functions is to apply a function repeatedly to items (or successive sublists) of given lists as arguments.

A black text with black letters

Description automatically generated with medium confidence

* **MAPCAR** 
  + Each intermediary result in a list
  + applied to:
    - CAR
    - CADR
    - CADDR
    - …
* **MAPLIST** 
  + Each intermediary result in a list
  + applied to
    - the whole lists
    - CDRs
    - CDDRs
    - …
* **MAPCAN**
  + Same as MAPCAR
  + BUT returns all intermediary results merged using NCONC
  + So the intermediary results must be LISTS
  + Otherwise 🡪 returns NIL
* **MAPCON**
  + Same as MAPLIST but results with NCONC
* **MAPC**

A white paper with black text

Description automatically generated