Lisp Forms

- Can write

  `CONS` of `t` and `nil`

  as

  `(CONS t nil)`

  $\boxed{\text{(CONS t nil)}}$ is called a *FORM*.

- In general: A *FORM* is

  – `(fn` $\mathtt{a}_1$ `...` $\mathtt{a}_n$`)`
  where `fn` is a function, and
  each $\mathtt{a}_i$ is a form

  – a *constant*
  (*e.g.*, `t`, `nil`, a number, . . .)

  – a *variable*.

Examples of Forms

t	nil
5	19

(CONS t nil)

(CONS t (CONS 5 nil))

(+ 3 (* 4 6))

(+ 4 (CONS 19 4))

Evaluation in Lisp

- *Lisp* based on *Evaluation*

  `EVAL` of a FORM is an S-expr.

Eg1: `CONS` of `t` and `nil` is `(t)`

$\approx$

`EVAL` of `(CONS t nil)` is

“`CONS`” applied to “`t`” and “`nil`”

$\leadsto$ `(t)`

- So

  `EVAL` of (fn a_1 ... a_n) is

  “fn” applied to “a_1”, . . . , “a_n”

  . . . but what of . . .

Evaluation in Lisp — II

Eg2: `EVAL` of `(CAR (CONS t nil))` is

“`CAR`” applied to <u>`EVAL`</u> of “`(CONS t nil)`”

`EVAL` of “`(CONS t nil)`” $\rightsquigarrow$ `(t)`

`EVAL` of “`(CAR` $\boxed{\texttt{(t)}}$`)`” $\rightsquigarrow$ `t`

Eg3: `EVAL` of `(FCTRIAL (+ 2 3))` is

“`FCTRIAL`” applied to <u>`EVAL`</u> of “`(+ 2 3)`”

`EVAL` of “`(+ 2 3)`” $\rightsquigarrow$ 5

`EVAL` of “`(FCTRIAL` $\boxed{5}$`)`” $\rightsquigarrow$ 120

Evaluation

- To find value of form: (fn a_1 ...a_n):

  1. EVALuate each of a_1 through a_n
     obtaining v_1 through v_n

  2. Apply fn to v_1 through v_n

- Can be arbitrarily recursive:
  EVAL of (+ 2 (* (- 9 3) 4))

What *Lisp*'s interpreter does:

```
→ (+ 2 3)
5

→ (CONS t NIL)
(t)
```

`EVAL` of Atoms

- "`EVAL`" maps each atom to its value.

- An *atom* may be
  a *constant* or a *variable*.

- Constants — "`t`", "`nil`", Numbers —
  evaluate to themselves.
  Value of nil is nil
  Value of 9 is 9

  ... means Value of `(CONS 9 nil)` is `(9)`
  as "`9`" $\leadsto$ `9` and "`nil`" $\leadsto$ `nil`

- Other atoms — variables —
  may *evaluate* to any s-expr.
  (A variable need not have a value.)

Evaluation of Variables

∗ Use of `SETQ`

After `(SETQ abc 5)`,

Value of	`abc`	is `5`	
Value of	`(+ abc 9)`	is `(+ 5 9)`	$\leadsto$ `14`
Value of	`(CONS abc nil)`	is `(CONS 5 nil)`	$\leadsto$ `(5)`

- Eg:

  value of `abc` could be `5`
  or `(x y)`
  or `abc`
  or `nil`
  or it could be *undefined*

- The value of an atom can be an atom; that atom may also have a value:

  E.g.,

value of	`abc`	could be	`x`
value of	`x`	could be	`tom`
value of	`tom`	could be	`(b c)`

Evaluation

$\rightarrow$ `(+ 3 4)`

`7`

$\rightarrow$ `(CONS 5 NIL))`

`(5)`

if abc's value is $\boxed{\texttt{5}}$

$\rightarrow$ `(CONS abc NIL))`

`(5)`

if also def's value is $\boxed{\texttt{(A)}}$

$\rightarrow$ `(CONS def (cons abc NIL)))`

`((A) 5)`

$\rightarrow$ `(CAR def)`

`A`