Logic and Functional Programming

Basic Elements of LISP

Dr. Cristian-Paul Bara

Computer Science Department
Faculty of Mathematics and Computer Science,
Babeş Bolyai University,
Cluj-Napoca, Romania

Table of Contents

LISP Functions

- (LIST e_1 e_2 ... e_n): list
- Forms the list of the arguments
  (LIST 'A 'B) = (A B)
  (LIST '(A B) 'C) = ((A B) C)
  (LIST 'A) = (A) $\leftrightarrow$ (CONS A NIL)
  (LIST '(A B C) NIL) = ((A B C) NIL)

LISP Functions

- (APPEND e_1 e_2 ... e_n): list
- Repeatedly calls CONS on the last two element and returns the resulting list
- If the arguments are lists it effectively concatenates them
- In CLISP the arguments cannot be atoms, except for the last one
- Memory inefficient, the first argument is always copied before being linked to the next

  (APPEND '(A B) '(C D)) = (A B C D)

  (APPEND '(A B) 'C) = (A B . C)

  (APPEND '(A B C) '()) = (A B C)

LISP Functions

- (ATOM e): T,NIL
- T if the argument is an atom or NIL otherwise
  (ATOM 'A) = T
  (ATOM A) = depends to what A evaluates to
  (ATOM '(A B C)) = NIL
  (ATOM NIL) = T

LISP Functions

- (LISTP e) : T, NIL
- T if the argument is a list or NIL otherwise
  (LISTP 'A) = NIL
  (LISTP A) = T A evaluates to a list
  (LISTP '(A B C)) = T
  (LISTP NIL) = T

LISP Functions

- (EQUAL e_1 e_2) : T, NIL
- T if the arguments have the same structure

  (EQUAL '(A B) '(A B)) = T

  (EQUAL '(A B) '(A (B))) = NIL

  (EQUAL 3.0 3.0) = T

  (EQUAL 8 8) = T

  (EQUAL 'a 'a) = T

LISP Functions

- (NULL e) : T, NIL - T is the argument evaluates to an ampty list or NIL atom, or NIL otherwise
- (NUMBERP e): T,NIL - T if the argument is a number, NIL otherwise
- (NOT e): T,NIL - T if the argument evaluates to NIL, NIL otherwise. Equivalent tu NULL
- (AND e_1 e_2 ... e_n): T,NIL - Evaluated left to right until the first NIL in which case it returns NIL, or the value of the last argument
- (OR e_1 e_2 ... e_n): T,NIL - Evaluated left to right until the first value other than NIL in which case it returns that value, or NIL otherwise

  (OR (cdr l) (car l)) will return (cdr l) if (cdr l) is not NIL or (car l) otherwise

  (or (cdr ()) (car '(3 4))) $\rightarrow$ 3

  (or (cdr '(1 2)) (car '(3 4))) $\rightarrow$ (2)

LISP Functions

- Arithmetic Operators:
  (+ n_1 n_2 ...); (- n_1 n_2 ...); (* n_1 n_2 ...);
  (/ n_1 n_2 ...); (MAX n_1 n_2 ...); (MIN n_1 n_2 ...)
- Relational Operators
  = for numbers only, <, <=, >, >=

LISP Functions

- (COND
  (<*conditional expression*> <*return expression*>)
  (<*conditional expression*> <*return expression*>)
  ...
  (t <*default return expression*>)
)
- (COND
  ((> x 5) (cons 'a '(b)))
  (t 'a)
)
- returns (a b) for x=7 and a for x=4

LISP Functions

- (DEFUN <*function name*> (<*arguments*>)
  <*symbolic expression*>
  <*symbolic expression*>
  ...
  <*symbolic expression*>
  <*symbolic expression to be returned*>
)
- (DEFUN MAX (X Y)
  (COND
  ((> X Y) X)
  (T Y)
)
)

LISP Functions

- DEFUN can redefine inbuilt functions
- (DEFUN f (l)
  (CAR l)
)
- (DEFUN f (l_1 l_2)
  (CDR l_2)
)
- In case a function is redefine with a different arity the last one is kept. (f '(1 2 3)) will throw an error while (f '(1 2) '(3 4)) will return (4)

LISP Functions

- What does this function return for (F 2 5) or (F 5 2)?
- (DEFUN F (X Y)

```
    (COND
        ((< X Y) X)
        (T Y)
    )
    Y
)
```

LISP Functions

- What does this function return for (F 2 5) or (F 5 2)?
- (DEFUN F (X Y)

```
    (COND
        ((< X Y) X)
        (T Y)
    )
    Y
)
```

- (F 2 5) $\rightarrow$ 5
- (F 5 2) $\rightarrow$ 2

How to Install

- Ubuntu: `$ sudo apt-get -y install clisp`
- Mac: `$ brew install clisp`
- Windows: Click here to download