

Lucrarea 1

Scurt istoric LISP

Limbajele de programare sunt de doua tipuri:

- procedurale (indica cum se rezolva o problema - pas cu pas)
- neprocedurale (indica subproblemele ce trebuie rezolvate).

LISP este un limbaj neprocedural, al doilea ca vechime dupa FORTRAN. A fost inventat de către [John McCarthy](#) în 1958. În 1965 - 1975 - se caută noi variante. La începutul anilor 1970 existau două dialecte: MacLisp și Interlisp. MacLisp s-a dezvoltat din Lisp 1.5 prin adăugarea variabilor speciale, și a tratării erorilor.

Interlisp a introdus mai multe concepte noi, construcția iterativă, preluate de MacLisp și mai târziu de CommonLisp. În 1981 se impune COMMON LISP, care este implementat în 1984 ca descrierea unei familii de limbaje.

În cadrul laboratorului de CFLP vom folosi pachetul XLISP-stat.

Salvați xisp-stat pentru Windows de la <ftp://ftp.stat.umn.edu/pub/xispstat/> versiunea cea mai recentă.

LISP este limbajul care se bazează pe programarea funcțională, conceptul de prelucrare a datelor fiind cunoscut sub numele de prelucrare simbolică.

O funcție LISP este un obiect matematic care întoarce o unică valoare pentru un set de valori de intrare. Funcțiile întorc valori atunci când sunt evaluate.

Sintaxa Lisp

Următoarele caractere au o semnificație specială în Lisp:

- (– o paranteză stângă marchează începutul unei liste;
-) – o paranteză dreaptă marchează sfârșitul unei liste;
- ' – un apostrof, urmat de o expresie e, 'e, reprezintă o scriere condensată pentru un apel (quote e); se mai folosește pentru a marca începutul unui mesaj ce urmează a fi tipărit
- ; – punct-virgula marchează începutul unui comentariu. El însuși împreună cu toate caracterele care urmează pînă la sfârșitul rîndului sînt ignorate;
- " – între o pereche de ghilimele se include un șir de caractere;

LISP este un limbaj pentru prelucrarea listelor. Procedurile și datele au aceeași structură, și anume o listă generalizată.

Elementele unei liste pot fi simboluri, ce definesc **ATOMII**, și alte **LISTE**, formând așa numita **expresie simbolică**.

Elementele unei liste sunt separate sintactic printr-un singur spațiu.

Implementările uzuale de Common Lisp sunt insensibile la forma caracterelor (minuscule sau majuscule). Intern însă, formele Lisp sînt reprezentate cu majuscule, de aceea formele rezultate în urma evaluărilor sunt redată în majuscule.

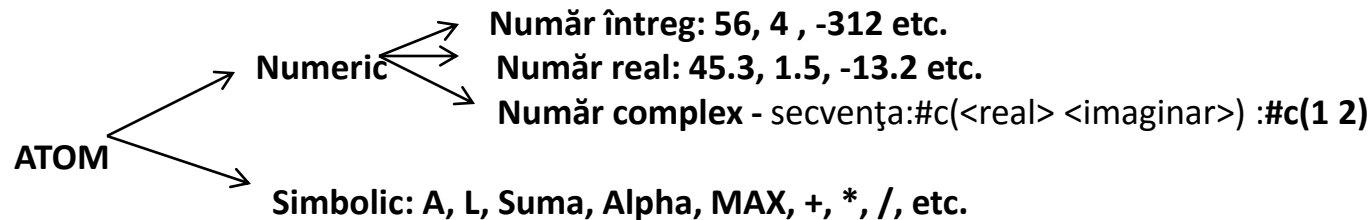
Exemple expresii simbolice:

(a alfa b (c d e) delta) – listă cu 5 elemente, 4 atomi simbolici și o listă cu 3 elemente atomi simbolici

(3 4 5 (6 7)) – listă cu 4 elemente, 3 atomi numerici și unul , listă cu 3 elemente atomi numerici

Tipuri de date elementare

Tipurile de date elementare în LISP sunt:



Orice combinație de litere și cifre poate constitui un nume de **simbol**. În plus, oricare dintre următoarele caractere poate interveni într-un nume de simbol: + - * / @ \$ % ^ & _ = < > ~

LISTA – expresie de forma: (elem1 elem 2 ... elem n)

Lista poate conține oricâte elemente de tip atom sau listă separate printr-un singur spațiu

LISP-ul ignoră spațiile în plus sau ENTER

Dacă toate parantezele deschise nu sunt închise de utilizator, LISP-ul așteaptă în continuare paranteză închisă. (ATENȚIE!!! Expresia nu poate fi evaluată)

Se pot pune mai multe paranteze dreapta decât stânga, LISP-ul le va ignora pe cele în plus.

EXEMPLE:

- () – lista vidă, notată și nil sau NIL;
- (a atom 3) – lista formată din 3 atomi, dintre care primii doi simbolici, iar al treilea numeric;
- (atom (a) elem (b 1) lista (c 1 2)) – lista formată din 6 elemente: primul, al treilea și al cincilea fiind atomi simbolici, iar al doilea, al patrulea și al șaselea – fiind la rândul lor liste cu unu, două, respectiv trei elemente, câte unul nenumeric, restul numerici.
- ((ATOM) Atom)))) – parantezele stânga în plus sunt ignorate, expresie simbolică corectă

Construcțiile limbajului

În LISP operăm cu următoarele categorii de obiecte:

- **Variabile**
- **Constante**
- **Date**
- **Funcții**
- **Macrouri**

Numai datele au tipuri, o variabilă în LISP nu are definit un tip. Ea poate primi ca valoare orice tip de dată.

Există trei moduri prin care o variabilă poate primi valori:

- **prin asignare**
- **prin legare**
- **prin listă de proprietăți**

Constantele sunt simboluri care au atașate valori ce nu pot fi modificate.

În Lisp nu există proceduri ci numai funcții, în sensul că orice rutină întoarce obligatoriu și o valoare.

Macro-urile sunt funcții care au un mecanism de evaluare special, în doi pași: expandarea și evaluarea propriu-zisă.

Evaluarea expresiilor

Expresiile Lisp sunt interpretate într-o buclă READ – EVAL – PRINT, în care are loc o fază de citire a expresiei de pe fluxul de intrare, urmată de evaluarea expresiei, și o ultimă fază de tipărire a rezultatului.

Eval – evaluează expresia simbolică introdusă

Orice expresie ce urmează a fi evaluată se prezintă pe un rând imediat după *prompter* (>), pentru ca pe rândul următor să apară rezultatul tipărit al evaluării.

[illegible]

- un atom simbolic care are o valoare predefinită (în sensul asignării sau al legării) se evaluează la această valoare.

- un atom numeric se evaluează la el însuși:

3

> a

Excepție fac simbolurile: **nil** care, fiind notația pentru lista vidă cât și pentru valoarea logică fals, se evaluează la el însuși , și **t** care se evaluează la valoare logică adevărat (true sau TRUE). Orice expresie diferită de nil este considerată a fi echivalentă logic cu true:

NIL

TRUE

Evaluarea expresiilor

- un atom simbolic care nu are o valoare predefinită, tentat a fi evaluat, va provoca un mesaj de eroare de genul UNBOUND-VARIABLE;
- o expresie simbolică prefixată cu apostrof (*quote*) se evaluează la ea însăși:
 - > 'alpha
alpha
 - > '3
3
 - > '(a 3 ())
(a 3 NIL)
- o listă care are primul element o forma LISP predefinită (operatori aritmetici, funcții matematice, primitive LISP) sau o funcție definită de utilizator, evaluează restul elementelor listei conform formei respective.

EXAMPLE:

```
> (+ 2 1)
3
>(+ (* 3 2) (- 8 2))
12
>(expt 3 2)
9
```

Probleme

1. Recunoașteți următoarele expresii și precizați tipul de date(atom sau listă):
((atom a (3) elem))
3
(3)
)(
(((a b c)
Atom
(atom elem))))
2. Precizați numărul de elemente ale următoarelor liste, și apoi tipul elementelor:
((1 3) 3.4 (a b c) ((atom)))
((a (b c)) d)
((((a))) ((b)) (c) d)
(a b cd)
3. Scrieți exemple de liste :
Cu 4 elemente atomi de tip numeric
Cu 3 elemente de tip listă
Cu 2 elemente atomi simbolici si 2 elemente de tip listă, fiecare cu câte 2 elemente atomi numerici