



THIS PAGE IS  
INTENTIONALLY  
LEFT BLANK.

Kazimir Majorinc

# REKURZIVNE FUNKCIJE NA SIMBOLIČKIM IZRAZIMA III.

Povijest Lispa 9.

Razmjena vještina  
Hacklab u mami  
20. listopada 2012.

# *eval.*

```
eval [e; a] = [  
atom [e] → assoc [e; a];  
atom [car [e]] → [  
eq [car [e]; QUOTE] → cadr [e];  
eq [car [e]; ATOM] → atom [eval [cadr [e]; a]];  
eq [car [e]; EQ] → [eval [cadr [e]; a] = eval [caddr [e]; a]];  
eq [car [e]; COND] → cvcon [cdr [e]; a];  
eq [car [e]; CAR] → car [eval [cadr [e]; a]];  
eq [car [e]; CDR] → cdr [eval [cadr [e]; a]];  
eq [car [e]; CONS] → cons [eval [cadr [e]; a]; eval [caddr [e];  
a]]; T → eval [cons [assoc [car [e]; a];  
evlis [cdr [e]; a]]; a]];  
eq [caar [e]; LABEL] → eval [cons [caddr [e]; cdr [e]];  
cons [list [cadar [e]; car [e]; a]];  
eq [caar [e]; LAMBDA] → eval [caddr [e];  
append [pair [cadar [e]; evlis [cdr [e]; a]; a]]]
```

1. 
$$\text{eval } [e; a] = [$$

$e$  je S-izraz koji se izračunava

$a$  je S-izraz koji sadrži vrijednosti varijabli, npr  $((X, A), (Y, (A, A)))$

$\text{atom } [e] \rightarrow \text{assoc } [e; a];$

2.

$e$  je atomarni izraz. U tom slučaju, funkcija *assoc* vraća odgovarajuću vrijednost koja je spremljena u  $a$ .

Primjerice,

$\text{assoc}[B; ((A, X), (B, XX))] = XX.$

3. `atom [car [e]] → [`

Grana koja se izvršava ako sam S-izraz `e` nije atomaran, ali prvi element izraza jest. Tj, `e` ima neki od oblika `(QUOTE, e0)`, `(ATOM, e0)`, `(EQ, e1, e2)`, `(COND, e1, ..., en)`, `(CAR, e0)`, `(CDR, e0)`, `(CONS, e1, e2)`.

Ne i `((LAMBDA, ...), ...)` ili `((LABEL, ...), ...)`.

$eq [car [e]; QUOTE] \rightarrow cadr [e];$

4.

Ako je  $e$  oblika  $(QUOTE, e0)$ , onda  $eval[(QUOTE, e0); a] = e0$ .

5.  $eq [car [e]; ATOM] \rightarrow atom [eval [cadr [e]; a]];$

Ako je  $e$  oblika  $(ATOM, e0)$  *eval* izračunava  $e0$  (uz iste vrijednosti varijabli) i primjenom S-funkcije *atom* provjerava je li rezultat atom.



6.  $eq [car [e]; EQ] \rightarrow [eval [cadr [e]; a] = eval [caddr [e]; a]];$

Ako je  $e$  oblika  $(EQ, e1, e2)$ , onda  $eval$  izračunava  $e1$  i  $e2$  (uz iste vrijednosti varijabli) i provjerava jesu li rezultati jednaki.

7.  $eq [car [e]; COND] \rightarrow evcon [cdr [e]; a];$

Ako je  $e$  oblika  $(COND, ((p1, e1), \dots, (pn, en)))$ , onda se poziva S-funkcija  $evcon[((p1, e1), \dots, (pn, en)); a]$ , posebno definirana za ovaj slučaj.

$$evcon[c; a] = [eval[caar[c]; a] \rightarrow eval[cadar[c]; a]; \\ \mathbf{T} \rightarrow evcon[cdr[c]; a]]$$

$evcon$  provjerava je li  $eval[p1; a] = \mathbf{T}$ .

Ako jest onda vraća  $eval[e1; a]$ .

Ako nije, onda poziva  $evcon[((p2, e2), \dots, (pn, en)); a]$

8.  $\text{eq} [\text{car} [e]; \text{CAR}] \rightarrow \text{car} [\text{eval} [\text{cadr} [e]; a]];$

Ako je  $e$  oblika  $(\text{CAR}, e_0)$  onda se izračunava  $e_0$  (uz vrijednost varijabli  $u$   $a$ ) i vraća prvi element rezultata.

9.  $\text{eq} [\text{car} [e]; \text{CDR}] \rightarrow \text{cdr} [\text{eval} [\text{cadr} [e]; a]];$

Ako je  $e$  oblika  $(\text{CDR}, e_0)$  onda se izračunava  $e_0$  (uz vrijednost varijabli  $u$   $a$ ) i vraća ostatak rezultata.

$eq [car [e]; CONS] \rightarrow cons [eval [cadr [e]; a]; eval [caddr [e];$

$a]];$

10.

Ako je  $e$  oblika  $(CONS, e1, e2)$  onda se izračunavaju  $e1$  i  $e2$  (uz vrijednosti varijabli u  $a$ ) i funkcija  $cons$  se primjenjuje na rezultate.

$T \rightarrow \text{eval} [\text{cons} [\text{assoc} [\text{car} [e]; a];$

$\text{evlis} [\text{cdr} [e]; a]]; a];$

11.

Zagonetni ogranak. Služi za izračunavanje izraza koji na prvom mjestu imaju ime funkcije koja je definirana u  $a$ . Na primjer,

$\text{eval}[(K, (\text{QUOTE}, X)); ((K, \text{ATOM}))]$

$\text{evlis}[(e_1, \dots, e_n); a] = (\text{eval}[e_1; a], \dots, \text{eval}[e_n; a])$

$\text{evlis}[(\text{ATOM}, X), (\text{ATOM}, (X))]; () = (T, F)$ .

Ovdje je suvišan!!!

$\text{eval}[(K, (\text{QUOTE}, X)); ((K, L), (L, M), (M, N), (N, \text{ATOM}))]$  funkcioniра.

eq [caar [e]; LABEL] → eval [cons [caddar [e]; cdr [e]];  
cons [list [cadar [e]; car [e]; a]];

12.

*eval*[( (LABEL, F, (LAMBDA, (...), ...)), ...)] ; a]

se svodi na

*eval*[( (LAMBDA, (...), ...)), ...] ;

*append*[(F, (LABEL, F, (LAMBDA, (...), ...)))] ; a]

$eq [caar [e]; LAMBDA] \rightarrow eval [caddar [e];$

13.  $append [pair [cadar [e]; evlis [cdr [e]; a]; a]]$

$eval[(LAMBDA, (v1, \dots, vn), e0), e1, \dots, en]; a]$

se svodi na

$eval[e0;$

$append[(v1, eval[e1; a]), \dots, (vn, eval[en, a])] ; a]$