



THIS PAGE IS  
INTENTIONALLY  
LEFT BLANK.

Kazimir Majorinc

# Od Zdravog razuma do Rekurzivnih funkcija.

Povijest Lispa 6.

t

Razmjena vještina

Hacklab u mami

29. rujna 2012.

MIT AIM-004. **John McCarthy**, *Symbol Manipulation Language - Revisions of the language*. Bez datuma. 22 strane.

**1. Zaštićena privremena pohrana.** Kako implementirati rekurzivne funkcije da različite instance međusobno ne brišu podatke? Tako da svaka funkcija spremi vrijednosti *registara* koje koristi na *stack*, izvrši se, i onda rekonstruira vrijednosti registara. (IBM 704 manual: “registar” označava i memorijske lokacije.)

**2. Revidirana verzija maplista.** Po AIM-001, primjerice, ako **L** ima vrijednost **(1,2)** onda

**maplist(L, J, J\*J+1) => (2, 5).**

Revidirani je oblik

**maplist(L,  $\lambda(J, J*J+1)$ ).**

Tek sada funkcija postaje vrijednost „prvog reda.“ Je li ovo početak funkcionalnog programiranja?

Kako je došlo do ove revizije?

Pogodnija je za implementaciju (?)



The version of maplist in memo 1 was written "maplist(L,J,f(J))" where J is a dummy variable which ranges over the address parts of the words in the list L and f(J) was an expression in J. This version had two serious defects. First, the location of the word in which J was stored was frequently needed. The second turned up when I tried to write the SAP program for maplist. The designation of J as the name of the indexing variable cannot conveniently be done in the calling sequence of maplist. Instead we do it in specifying the function f using the Church notation for functional abstraction if necessary. In addition to the above mentioned defects the old version was ambiguous in that it did not say how words of the three types should be treated.

### 3-4. Primjeri kompajliranih funkcija u IBM 704 assembleru.

#### 3. Some Examples of SAP Language Programs

We give some examples of the way in which some of the programs described earlier in **LISP1** (List Processor) have been coded by hand in SAP. These examples are given so that

Je li ovo prvo spominjanje riječi Lisp? **Stoyan** kaže da je spomenut i u AIM-003.

## 5. Dodatne funkcije.

**select**, varijanta uvjetnog grananja, switch-case u C.

**list(i1,...,in)** ne može se definirati u jeziku zbog varijabilnog broja argumenata.

**eq1** i **cop1**, uspoređuju i kopiraju liste, bez podlista. Općenitiji oblik je već definiran.

**search, subst, error, pair...**

**apply(L, f)**, definiran samo “ako prvi element **f** je **subfun**.” Kakav **subfun**? Subfun nije prije spomenut, a po mišljenju **Stoyana**, to je **lambda**.

Primjerice, ako **L** ima vrijednost **(1, 2)** onda

$$\mathbf{apply(L, \lambda(x, y, x+y))} \Rightarrow 3.$$

U memou je i definicija funkcije **apply** uz pomoć prethodno definiranih funkcija, posebno **subst**.



MIT AIM-005. **Nathaniel Rochester**, *Symbol Manipulation Language - bez naslova*. 20. studeni **1958**, 28 strana.



**Nathaniel Rochester (1919 - 2001)**, dizajnirao IBM 701 i “Simbolički assembler”, istraživao umjetnu inteligenciju dok IBM nije zaključio da AI ideje plaše kupce i smanjuju prodaju računala.

**Program za deriviranje.** Opisan u puno detalja. Najteži dio deriviranje mnogostrukih produkta ( $E_1 * E_2 * \dots * E_n$ ). Uključuje simplifikaciju izraza. Nekoliko novih ideja:

**C...R**

`car(car(J)) = caar(J)`

`car(cdr(J)) = cadr(J)`

`car(cdr(car(cdr(car(cdr(J)))))) = caa2cadr(J) etc.`

## Formatiranje koda:

$(A \rightarrow B, B \rightarrow (C \rightarrow D, E \rightarrow F), G \rightarrow H)$  is written  
A  $\rightarrow$  B,  
B  $\rightarrow$  (C  $\rightarrow$  D,  
    E  $\rightarrow$  F),  
G  $\rightarrow$  H)

## Funkcija Replace:

**Replace(L, M)** umjesto **L = M**.

MIT AIM-006. **Steve Russell**, *Writing and debugging programs*. Bez datuma. 7 strana.



**Steve “Slug” Russell, (1937-),** glavni programer projekta, autor prve video igre.

Valja napisati posebne programe za testiranje, i to:

- (1) Za svaki specijalan slučaj inputa jedan program.
- (2) Za svaki specijalan slučaj izračunavanja jedan program
- (3) Nekoliko programa za tipične slučajeve.

“Print early and ofthen”.

Prvo isprobaj trivijalne testove!

Koristi paket koji smo razvili za debugiranje.

Čitač kartica je pokvaren pa se slova I, R i Z se ne smiju koristiti (istina, samo u 73. koloni.)

## MIT AIM-007. John McCarthy, *Notes on the compiler*. 2 strane. Nema datuma.

Kompajler je definiran kao funkcija **compute** koja se poziva sa **compute(L,C)**, gdje je **L** izvorni program, **C** mjesto na koje se sprema rezultat: program u simboličkom assembleru. Funkcija compute je rekurzivna.

Uputa je djelomična i nerazumljiva bez poznavanja IBM 704 assemblera, te vjerojatno lokalnih standarda.



Nakon ovog memoa dolazi do jaza od tri mjeseca.

Slijedeći memo, AIM-008 datiran je tek 13. ožujka **1959**. AIM-008 ima naslov “*Recursive functions of symbolic expressions and their computation by machine.*”

Slijedi niz memoa, ispravaka i drugih dokumenata sa istim naslovom, među kojima je i prvi objavljeni članak o Lispu, **1960**. Vjerojatno najvažniji članak o Lispu. Sve bitno što je dosad rečeno i nešto novog (**QUOTE**), ali ljepše izrečeno.