



THIS PAGE IS
INTENTIONALLY
LEFT BLANK.

Kazimir Majorinc

REKURZIVNE FUNKCIJE NA SIMBOLIČKIM IZRAZIMA IV.

Povijest Lispa 10.



Razmjena vještina
Hacklab u mami
27. listopada 2012.

SYSTEM ZA PROGRAMIRANJE LISPOM.

S-izrazi su reprezentirani *strukturuom liste*.

Ista podstruktura se može pojavljivati više puta unutar strukture liste.

Ciklusi nisu dozvoljeni.

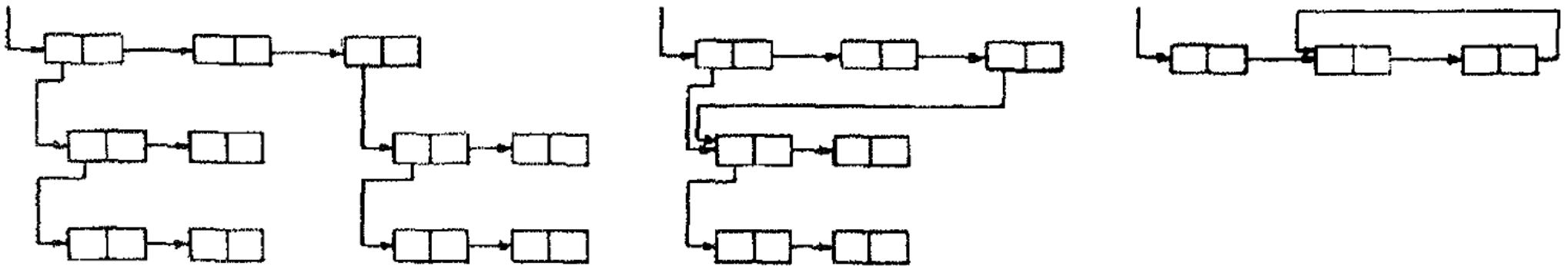


FIG. 1

S-izrazi u kojima se podizrazi pojavljuju više puta mogu biti reprezentirani na više različitih načina. Primjerice, $((A \cdot B) (A \cdot B))$ može biti reprezentiran kao

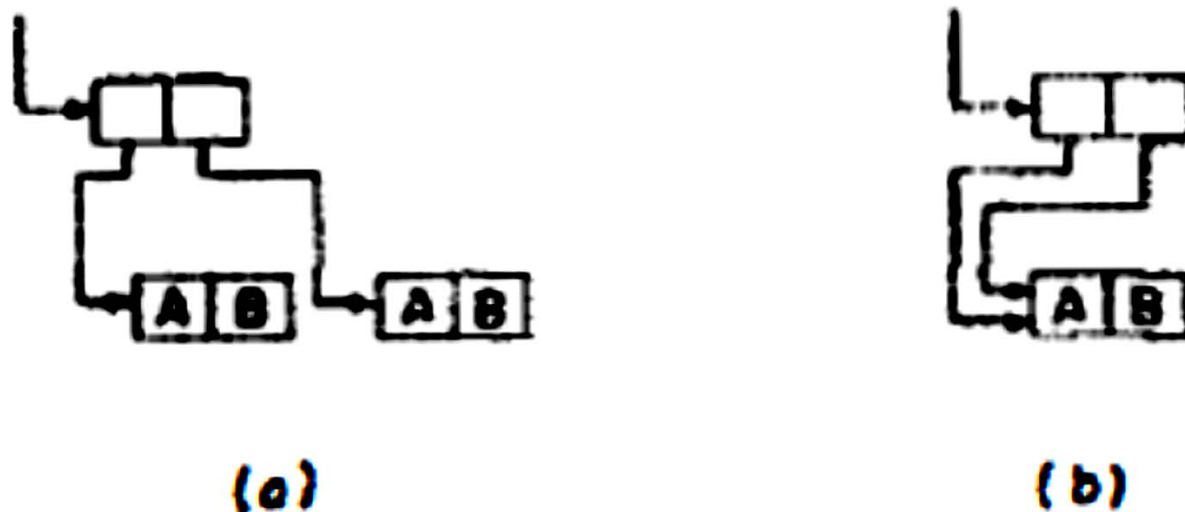


FIG. 3

Programi ne mogu raspoznati razliku između to dvoje. Razlika može postojati u prostoru i vremenu potrebnom za rad. Cirkularne liste bi mogle biti korisne za predstavljanje rekurzivnih funkcija, ali teškoće u ispisu

LISTE ASOCIJACIJA

Simboli imaju liste asocijacija u kojima se nalazi sve što želimo, je li, asociirati sa simbolom. Ime za ispis; vrijednost simbola; adresu potprograma u mašinskom jeziku.

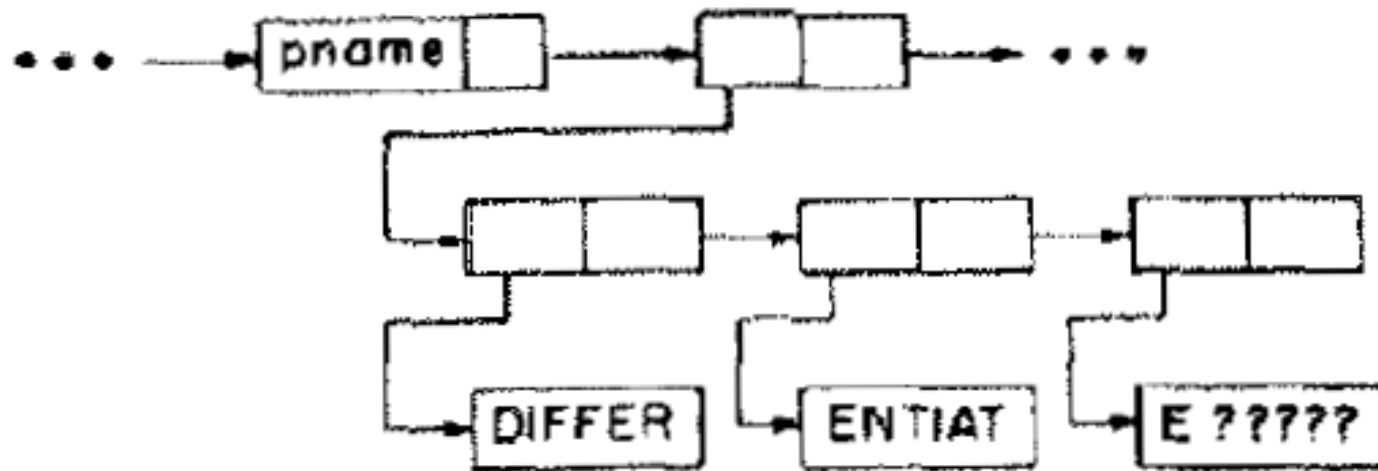


FIG. 4

Svaki simbol ima samo jednu listu asocijacija. S-funkcija *eq* uspoređuje lokacije liste asocijacija.

LISTA SLOBODNIH RIJEČI

Slobodna memorije (oko 15 000 riječi) je organizirana u listu; Riječ nazvana „FREE“ sadrži adresu prve slobodne riječi. Kad program zatreba slobodnu riječ, uzima je s početka liste, a adresa druge slobodne riječi upisuje se u FREE. Korisnik ne mora skrbiti o programiranju povratka memorije u listu FREE. Taj je povratak automatski.

Ništa se ne zbiva dok program ne potroši slobodnu memoriju. Kad je slobodna riječ potrebna a na listi slobodnih riječi je nema, počinje „reclamation cycle“.

Program nalazi sve riječi kojima korisnik može pristupiti putem „CAR-CDR lanca“ i postavlja njihov predznak na „negativan“. Ako pri tom naiđe na riječ sa već negativnim predznakom, pretpostavlja da je ta riječ već dohvaćena.

Nakon toga, program prolazi kroz cijelu memoriju dostupnu korisniku, i riječi sa pozitivnim predznakom vraća natrag na listu slobodnih riječi. Konačno, sve negativne predznake ponovo postavlja na pozitivne.

LINEARNI LISP

Definicija L-izraza.

1. individualni znakovi (a takvih je konačan broj) su L-izrazi.
2. bilo koji niz individualnih znakova je L-izraz. Posebno, prazni niz znakova Λ je L-izraz.

Tri elementarne funkcije na L-izrazima su:

first[**ABC**]=**A**. *first*[Λ] nije definiran.

rest[**ABC**]=**BC**. *rest*[Λ] nije definiran.

combine[**A**;**BC**]=**ABC**.

Tri predikata:

char[*x*] - *x* je jedan znak

null[*x*] - *x* je Λ

$x = y$.

Prednosti linearnog Lispa su da je općenitiji i da koristi jednostavnije strukture podataka. Nedostatak je zahtjevnost izdvajanja podizraza. Zato bi u praksi bili korišteni S-izrazi. Podršku za S-izraze u linearnom Lispu nije teško implementirati, ali se onda gubi u efikasnosti u odnosu na standardni Lisp.

OD PROGRAMA DO S-FUNKCIJA

Stanje svakog programa određeno je vrijednošću svih varijabli u programu. U svakom trenutku, te se vrijednosti mogu složiti u jedan vektor.

Postoji funkcija f definiran na svim takvim vektorima da $f[X] = Y$ ako i samo ako stanje Y slijedi iz stanja X .

