THIS PAGE IS INTENTIONALLY LEFT BLANK.

Kazimir Majorinc

Rekurzivne funkcije na simboličkim izrazima.

Povijest Lispa 7.

X

Razmjena vještina
Hacklab u mami
6. listopada 2012.

John McCarthy, *Recursive functions of symbolic expressions and their computation by machine. Part I.* Communications of the ACM, 3(4), 1960, 184-195.

Nakon AIM-007, koji je napisan **1958**, pauza od tri mjeseca o kojoj nema dokumenata. AIM-008 iz ožujka **1959** ima gornji naslov. Niz ispravaka i internih dokumenata sa istim naslovom. McCarthy postepeno gubi kontrolu nad jezikom, tj. ljudi ubacuju svoje ideje i rade brže od njega.

1. Popis uobičajenih metoda za izgradnju matematičkih izraza i posebno, definicija funkcija. Npr. $\lambda((x,y),y\vee x)$ umjesto $(x,y)\mapsto y\vee x$.

2. Definiraju se simbolički izrazi (S-izrazi), `A`, `B` ... `Z`, `AA`, `AB`, ..., `(A,A)`,`(A,B)`, ...,`(A,(A,A))`...`(A,B,C)`. Logičke formule se mogu zapisati u tom obliku.`((FORALL,X),((X,^,2),>,0))`

3. Koristeći metode iz 1, definiraju se funkcije na S-izrazima. No, svaka definicija funkcije je i sama izraz; meta-izraz, M-izraz. Mala slova, uglate zagrade i točka-zarezi.

4. Pokazuje se kako prevesti meta-izraze u simboličke-izraze. Npr. $\lambda[[x;y];y\vee x]$ se prevodi u `(LAMBDA,(X,Y),(OR,Y,X))`

5. Definira se univerzalna funkcija *apply*. Npr.

f[`X`;`Y`] = *apply*[prijevod *f*;`(X,Y)`]

1. Funkcije i definicije funkcija.

a. Funkcije su parcijalne; nisu definirane na cijeloj domeni.

b. Iskazni izrazi. T i F. Izrazi dobijeni korištenjem $\wedge$ $\vee$ ~.

c. Uvjetni izrazi. $(p_1 \rightarrow e_1, p_2 \rightarrow e_2, \ldots, p_n \rightarrow e_n)$

d. Rekurzivne funkcije, $n! = (n = 0 \rightarrow 1, \mathrm{T} \rightarrow n \cdot (n\text{-}1)!)$

e. Od forme do funkcije. Npr. od izraza x^2+y, možemo konstruirati funkciju $\lambda((x,y),x^2+y)$

f. Definicija rekurzivne funkcije. Npr

$$label(fact,n,(n = 0 \rightarrow 1, \mathrm{T} \rightarrow n \cdot fact(n\text{-}1)))$$

2. Klasa simboličkih izraza.

Simbolički izrazi su nizovi znakova.

(1) Svaki konačan niz slova brojeva i razmaka je simbol. Svi simboli su simbolički izrazi. Primjerice, `A`, `ABA`, `APPLE PIE`

(2) Ako su e_1 i e_2 simbolički izrazi, onda je i `(`e_1`.`e_2`)` simbolički izraz. `(A.B)`, `((A.B).(B.A))` itd.

(3) Izraz `(`e_1`,`e_2`,`...`,`e_n`)` je "pokrata" za `(`e_1`.(`e_2`.(`...`(`e_n`.NIL)`...`)))`. Posebno, `()` je pokrata za NIL (iako to McCarthy ne kaže eksplicitno).

3. S-funkcije.

Definiraju se iterativno. Prvo, konačni broj atomarnih funkcija. Zatim, sve koje se mogu izgraditi osnovnim operacijama nad prethodno definiranim funkcijama.

(1). Atomarne funkcije:

atom - vrijednost je **T** ako je argument simbol, **F** inače.
eq - vrijednost **T** ako su obadva argumenta isti simbol. **F** inače.
car - vrijednost prvi element argumenta **(**e_1**.**e_2**)**.
cdr - vrijednost drugi element argumenta **(**e_1**.**e_2**)**.
cons[e_1;e_2] = **(**e_1**.**e_2**)**.

(2) Funkcije koje se mogu dobiti iz prethodno definiranih funkcija korištenjem metoda opisanih u 1.

Primjer: funkcija *ff*

Vrijednost *ff*[*x*] je prvi atomarni simbol u S-izrazu, ignorirajući zagrade.

Primjerice, *ff*[`((A.B).C)`]=`A`

$$ff[x]=[atom[x]\rightarrow x,\ {\sim}atom[x] \rightarrow ff[car[x]]]$$

ili

$$label[ff;x;[atom[x]\rightarrow x;\ {\sim}atom[x] \rightarrow ff[car[x]]]].$$