This page is intentionally left blank.

Kazimir Majorinc

Temelj za matematičku teoriju izračunavanja (II)

Povijest Lispa 19.

Razmjena vještina
Hacklab u mami
26. siječnja 2013.

Kako se riješiti rekurzivnih definicija?

U Lispu, funkcije se definiraju izrazima poput

$$f[x;y]=x+y$$

Tako definirane funkciji se mogu zamijeniti lambda-izrazima, npr.

$$\lambda[[x;y];x+y].$$

Ali, ako su funkcije rekurzivne, primjerice

$$fact[n]=[n = 0 \rightarrow 1; T \rightarrow fact(n-1)*n]$$

njih nije moguće zamijeniti lambda izrazima jer u

$$lambda[[n]; n = 0 \rightarrow 1; T \rightarrow fact(n-1)*n]$$

nije jasno što znači fact. Umjesto toga, potreban je label-izraz

$$label[fact; lambda[[n]; n = 0 \rightarrow 1; T \rightarrow fact(n-1)*n]].$$

```
(define (fact m) (cond ((= m 0) 1)   ; rekurzivna
                       (true (* m (fact (- m 1))))))

; Brigo moja pređi na drugoga

(define (parazit n fi) (cond ((= n 0) 1) ; nije rekurzivna
                             (true (* n (fi (- n 1))))))

(println (parazit 5 fact)) ; radi

(println (parazit 0 dummy))  ; radi
;(println (parazit 1 dummy)) ; ne radi

(println (parazit 0 parazit))  ; radi
(println (parazit 1 parazit))  ; isto radi
;(println (parazit 2 parazit)) ; više ne radi
```

```
(define (fact m) (cond ((= m 0) 1)   ; rekurzivna
                       (true (* m (fact (- m 1))))))

(define (parazit n fi) (cond ((= n 0) 1) ;nije rekurzivna
                             (true (* n (fi (- n 1) fi)))))

(println (parazit 5 parazit)) ; radi

(define (fact n) (parazit n parazit))
(exit)
```

Neka je f rekurzivna funkcija definirana sa

$$f(x)=E(x,f)$$

gdje je E neki izraz u kojem se pojavljuju x i f. Ta se funkcija može zamijeniti label-izrazom. Definirajmo funkciju

$$p(x,fi) = E(x,\lambda((x),fi(x,fi))).$$

Funkcija p nije rekurzivna pa se može definirati lambda-izrazom, u ovom slučaju $\lambda((x,fi),E(x,\lambda((x),fi(x,fi)))$. Posebno, vrijedi jednakost

$$p(x,p)=E(x,\lambda((x),p(x,p))).$$

Definirajmo

$$f2(x)=p(x,p).$$

Niti f2 nije rekurzivno definirana i vrijedi $f2 = \lambda((x),p(x,p))$. Vrijedi

$$f2(x) = p(x,p) = E(x,\lambda((x),p(x,p)) = E(x,f2).$$

Dakle, f2 definirana isključivo lambda-izrazima zadovoljava polaznu jednakost.

Neizračunljive funkcije.

Osim lambda i label izraza, McCarthy razvija nove izraze kojima se mogu definirati funkcije. Neka je E izraz koji može imati vrijednosti T i F, i koji sadrži varijablu x. Definiramo novu formu

$$\forall((x),E) = \begin{cases} T & \text{ako } E \text{ ima vrijednost } T \text{ za sve } x \\ F & \text{ako } E \text{ ima vrijednost } F \text{ za bar jedan } x \\ \text{nedefiniran} & \text{inače} \end{cases}.$$

Uz pomoć te forme mogu se definirati funkcije koje nisu S-funkcije, tj. ne mogu se definirati u Lispu. Zašto? Zato što je $\forall((x)E)=F$ definirana čak i u slučaju da se za neki x izraz E izračunava do u beskonačnost.

Višeznačne funkcije.

Nisu uopće funkcije.

$$less(n)=amb(n-1,less(n-1)).$$

Višeznačne funkcije mogu se koristiti za definiranje drugih funkcija.

$$ult(n)=(n=0 \rightarrow 0,\ T \rightarrow ult(less(n)))$$

Tada vrijedi

$$\forall((n),ult(n)=0)=T.$$

Apstraktna definicija S-izraza.

Originalno, S-izrazi su nizovi znakova; npr. `((A.B).C)`.

Liste, poput `(A,B,C)` su pokrate, npr. `(A,(B,(C,NIL)`.

McCarthy definira S-izraze kao uređene n-torke nad nekim osnovnim skupom. Neka je *A* skup simbola {`A`, `B`, `C`, `AA` ... }. Tada je,primjerice,

$$((\texttt{A}\bullet\texttt{B})\bullet\texttt{C}) \in ((A \times A) \times A).$$

No, ((`A`•`B`)•`C`) je toliko sličan `((A.B).C)` da i njega - i sve slične - možemo smatrati S-izrazom. Skup svih S-izraza nad *A* se označava *sexp*(*A*).

$$sexp(A)=\{\Lambda\} \cup A \cup sexp(A) \times sexp(A)$$

Skup svih listi nad A je skup svih S-izraza oblika

$$(a_1,\ldots,a_n) = (a_1 \bullet (a_2 \bullet \ldots (a_n \bullet \Lambda))).$$

Taj skup je rješenje jednadžbe

$$seq(A) = \{\Lambda\} \cup A \times seq(A).$$

Na *sexp*(*A*) i *seq*(*A*) se mogu definirati uobičajene Lisp funkcije poput car, cdr, cons, atom, eq. Sve ostale definicije (eval, apply itd.) mogu ostati iste kao što su i bile.

Martin Davis u *Journal of Symbolic Logic* napisao da „nije uvjeren“ da iznesene ideje imaju teorijski značaj.