



THIS PAGE IS
INTENTIONALLY
LEFT BLANK.

Kazimir Majorinc

Imperativni aplikativni izrazi
i hibridi funkcija i labela

Povijest Lispa 31.



Razmjena vještina
Hacklab u mami
1. lipnja 2013.

Imperativni aplikativni izraz (IAE) je

1) identifikator

2) lambda-izraz koji se sastoji od vezane varijable i tijela koji je IAE

3) assigner - koji se sastoji od lijeve strane (lhs) koja je IAE i desne strane (rhs), isto IAE, npr. $lhs := rhs$

4) kombinacija - koja se sastoji od operatora i operanda.

Izračunavanje assignera

Bilo koji niz pravila koji

$(S, (\text{varijabla}, \dots):E, (\text{varijabla} := \text{izraz}):C, D)$

prevodi u

$(\text{izračunati izraz}: S, (\text{varijabla}, \text{izračunati izraz}):E, C, D)$

$([], e, [x], ()) \rightarrow \dots \rightarrow ([v], e, [], ())$.

Labele i procedure u Algolu

Primjećeno je da se simbol **go to** u Algolu 60 može odbaciti, te da se labele mogu tretirati kao procedure bez parametara, iz kojih se uvijek izlazi uz pomoć **go to** naredbe.

Ta analogija omogućuje da, u principu, labele i procedure imaju isti tip.

```
procedure P;  
  if p then go to M
```

```
...
```

```
L :  
  f(P) ... f(L)
```

Problem

```
begin
  S1;
  S2;
  L1: S3
  S4
  L2: L3: S5
  S6:
  L4: S7
  S8
end
L5;
...
```

```
begin
  procedure L1
    begin S3; S4; L2 end
  procedure L2; L3
  procedure L3;
    begin S5; S6; L4 end
  procedure L4:
    begin S7; S8; L5 end
  S1
  S2
  L1
end
L5; ...
```

Uvodi se **programpoint** deklaracija. U slučaju **prirodnog izlaska iz programpointa**, izlazi se ne samo iz tijela **programpointa**, nego i iz bloka u kojem je **programpoint** deklariran.

begin

programpoint L1
begin: S3; S4; L2 end

programpoint L2; L3

programpoint L3;
begin S5; S6; L4 end

programpoint L4:
begin S7; S8 end

S1

S2

L1

end

Programsko zatvorenje u SECD

programsko zatvorenje se sastoji od

- (1) tijela - koji je funkcija (ili funkcijsko zatvorenje)
- (2) dump-dijela - koji je SECD na mjestu na kojem je programsko zatvorenje definirano

J - operator koji zatvara funkciju u programsko zatvorenje. J je primitivni operator, ne da se definirati uz pomoć ničega prethodno definiranog (jer ima „uvid“ u cijeli SECD stroj)

Ako se u programu pojavi poziv programskog zatvorenja L, onda se

(1) pokuša primjeniti L na argumente

(2) ako nema „neprirodnog izlaska“ pozivom na neko drugo programsko zatvorenje, onda se na temelju dumpa rekonstruira stanje SECD stroja na mjestu na kojem je L bio definiran (a ne pozvan) i ta funkcija u kojoj je SECD bio definiran završava (sa istom vrijednošću kao i L).

Implementacija složena.