

Izračunavanje na riječima

Rečnik - to je čitav svijet poređan po azbučnom redosljedu.

Anatol Frans

5. Simulacija S_n u T

Teorema 5.1. Ako je $f(x_1, \dots, x_m)$ parcijalno izračunljiva u S_n , tada postoji Post-Tjuringov program koji striktno izračunava f .

Neka je Π program u S_n koji izračunava f . Pretpostavićemo da program Π , pored varijabli X_1, \dots, X_m i Y , koristi i lokalne varijable Z_1, \dots, Z_k . Tako program Π koristi ukupno $m+k+1$ varijabli: $X_1, \dots, X_m, Z_1, \dots, Z_k, Y$. Stavićemo da je $l = m+k+1$, a sve varijable, u navedenom redosljedu, prikazati kao niz V_1, \dots, V_l .

Konstruisaćemo Post-Tjuringov program Δ koji simulira program Π .

Najprije, definišimo početnu konfiguraciju trake kao:

$$B \ x_1 \ B \ x_2 \ B \ \dots \ B \ x_m \ B \ z_1 \ \dots \ B \ z_k \ B \ y$$

gdje su $x_1, x_2, \dots, x_m, z_1, \dots, z_k, y$ trenutne vrijednosti varijabli:

$$X_1, X_2, \dots, X_m, Z_1, \dots, Z_k, Y.$$

Sada ćemo konstruisati nekoliko makro instrukcija jezika T koje će nam olakšati posao.

Makro instrukcija:

$$\text{GOTO } L$$

dobija se programom

$$\text{IF } s_1 \ \text{GOTO } L$$

$$\text{IF } s_2 \ \text{GOTO } L$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$\text{IF } s_n \ \text{GOTO } L$$

Makro instrukcija:

RIGHT TO NEXT BLANK

dobija se programom

```
[A]  RIGHT
      IF B GOTO E
      GOTO A
```

Slično, makro instrukcija:

LEFT TO NEXT BLANK

se dobija programom:

```
[A]  LEFT
      IF B GOTO E
      GOTO A
```

Makro instrukcija:

MOVE BLOCK RIGHT

se dobija programom:

```
[C]  LEFT
      IF s0 GOTO A10
      IF s1 GOTO A1
      .
      .
      .
      IF sn GOTO An
[Ai] RIGHT
      PRINT A0           }i = 1,2,..., n  puta
      LEFT
      GOTO C
[A0]      RIGHT
      PRINT B
      LEFT
```

Efekat instrukcije MOVE BLOCK RIGHT je da od početne konfiguracije

ByB

formira konfiguraciju

BByB

odnosno da riječ y pomjeri udesno za jedno polje.

Makro instrukcija:

ERASE A BLOCK

se dobija programom:

```
[A]  RIGHT
      IF B GOTO E
      PRINT B
      GOTO A
```

Ovaj program, najprije, pomjeri glavu za čitanje udesno, a zatim "briše" redom sva polja dok ne naiđe na prvi blano znak, kada završava rad.

Uvešćemo konvenciju da brojem ≥ 0 u zagradi, na kraju makro instrukcije, označimo broj ponavljanja instrukcije. Na primjer,

RIGHT TO NEXT BLANK [3]

je skraćeno napisan programski segment:

```
RIGHT TO NEXT BLANK
RIGHT TO NEXT BLANK
RIGHT TO NEXT BLANK
```

Sada smo spremni da pokažemo kako se mogu simulirati sve tri instrukcije jezika S_n Post-Tjuringovim programom.

Počecemo sa instrukcijom oblika

$$V_j \leftarrow s_i V_j$$

Da bi ispred varijable V_j postavili simbol s_j , najprije sve varijable V_j, \dots, V_l moraju biti pomjerene za jedno polje udesno, da se napravi mjesto za s_j . Kada se s_j postavi na svoje mjesto, potrebno je glavu za čitanje vratiti na prvi blanko znak ispred V_1 , kako bi se ispravno nastavilo sa programom. To i radi sljedeći program:

```
RIGHT TO NEXT BLANK [l]
MOVE BLOCK RIGHT [l-j+1]
RIGHT
PRINT s_j
LEFT TO NEXT BLANK [j]
```

Sada moramo simulirati

$$V_j \leftarrow V_j^-$$

Ovdje nastaje komplikacija kada je V_j nula riječ, jer tada želimo da sve ostavimo nepromijenjeno. Zato se najprije pomjeramo do prvog blanko znaka desno od V_j . Sada, vraćajući se za jedno mjesto ulijevo možemo da ispitamo da li je V_j nula riječ. Evo programa.

```

RIGHT TO NEXT BLANK      [j]
LEFT
IF B GOTO C
MOVE BLOCK RIGHT        [j]
RIGHT
GOTO E
[C] LEFT TO NEXT BLANK   [j-1]
```

Zadnja instrukcija jezika S_n koju treba simulirati je

IF V_j ENDS s_j GOTO L

što se postiže sljedećim programom:

```

RIGHT TO NEXT BLANK      [j]
LEFT
IF  $s_j$  GOTO C
GOTO D
[C] LEFT TO NEXT BLANK   [j]
GOTO L
[D] RIGHT
LEFT TO NEXT BLANK      [j]
```

Ovim smo završili simulaciju S_n u T. Tako, svaki dati program Π u jeziku S_n možemo prevesti (kompilirati) u jezik T. Kada se prevedeni program izvrši na traci će ostati zapisano:

...B B B x_1 B...B x_m B z_1 B...B z_k B y B B B...

Međutim ako želimo da na kraju ostane smo izlaz, sve što treba da uradimo je da na kraj programa dodamo makro instrukciju:

ERASE A BLOCK [l-1]

Poslije ove naredbe konfiguracija je sljedeća:

...B B B B y B B B...

Zato kažemo da naša simulacija striktno izračunava funkciju f .
Ovim je teorema 5.1 dokazana.