GLAVA 4

Univerzalni program

Nesavršenstvo nije negacija savršenstva.
To se samo cjelina iskazuje u djelovima,
to se beskraj otkriva u granicama.
Rabindranat Tagore

4. Univerzalnost

Za svako n > 0 definišimo

$$\Phi^{(n)}(x_1,...,x_n,y) = \Psi_{\Pi}^{(n)}(x_1,...,x_n), \text{ gdje je } \#(\Pi) = y.$$

Jedna od ključnih teorema u teoriji izračunljivosti je

Teorema 4.1 (**T**eorema univerzalnosti) Za svako n > 0 funkcija $\Phi^{(n)}(x_1,...,x_n,y)$ je parcijalno izračunljiva.

Dokazaćemo ovu teoremu tako što ćemo, za svako n > 0, konstruisati program U_n koji izračunava $\Phi^{(n)}$. Drugim riječima imaćemo za svako n > 0:

$$\Psi_{U_n}^{(n+1)}(x_1,...,x_n,x_{n+1}) = \Phi^{(n)}(x_1,...,x_n,x_{n+1}) .$$

Program U_n zvaćemo **univerzalnim programom**.

Na primjer, program U_1 može biti korisćen za izračunavanje bilo koje parcijalno izračunljive funkcije sa jednom varijablom. Tako ako se f(x) izračunava programom Π i ako je y = #(Π), tada je f(x)=$\Phi^{(1)}$(x,y). Program U_n će raditi slično interpreteru. Mora voditi računa o trenutnom stanju programa i, "dekodiranjem" broja programa koji trenutno izvršava, odlučivati sta da radi i to raditi.

Pri pisanju programa U_n slobodno ćemo koristiti makro naredebe za sve funkcije za koje smo utvrdili da su primitivo rekurzivne.

Za potrebe "pamćenja" stanja programa, odnosno pamćenja vrijednosti promjenljivih tokom izvršavanja rogram U_n , koristićemo ponovo Gedelove

brojeve. Kako se u svakom programu koristi konačan skup varijabli, to se uvijek može odrediti Gedelov broj $[(a_1,...,a_m)]$, gdje je a_m vrijednost varijable sa najvećim rednim brojem u listi varijabli takva da je $a_m \neq 0$. Sve varijable koje se ne nalaze u listi Gedelovih vrijednosti imaju vrijednost 0.

Tako će, na primjer, stanje programa

$$Y = 0,\ X_1=2,\ X_2=1$$

biti kodirano sa $[0, 2, 0, 1] = 63$.

Lako je uočiti da sve ulazne (X) varijable imaju paran indeks, što će olaksati kodiranje.

Pri pisanju U_n programa koristićemo sljedeće oznake:

- K će biti broj sljedeće instrukcije koju treba izvršiti (programski brojač)
- S će sadržati stanje varijabli kao gore opisan Gedelov broj.

Sada ćemo nastaviti sa konstrukcijom programa U_n za izračunavanje funkcije:

$$Y = \Phi^{(n)}(x_1,...,x_n,x_{n+1}) .$$

Program ćemo razvijati "parče po parče" (modularno) , a na kraju ćemo sve djelove sakupiti u jednu cjelinu.

Počnimo sa:

$$Z \leftarrow X_{n+1} + 1$$

$$S \leftarrow \prod_{i=1}^{n} (p_{2i})^{X_i}$$

$$K \leftarrow 1$$

Kako je:

$X_{n+1} = \#(\Pi)$,

a Π se sastoji od instrukcija $I_1,...,I_m$,

onda Z dobija vrijednost $[\#(I_1),...,\#(I_m)]$.

S dobija početnu vrijednost $[0, X_1, 0, X_2, 0,..., X_n]$ čime se u početnom stanju zadaju ulazne vrijednosti varijablama, a sve ostale postavljaju na 0.

Programskom brojaču K se daje početna vrijednost 1 da izračunavanje krene sa prvom instrukcijom.

Slijedi:

[C] IF K = D(Z) + 1 $\veebar$ K = 0 GOTO F

Ovo je uslov za završetak rada programa: programski brojač je stigao do zadnje instrukcije, ili je postao jednak 0, (pod kojim uslovom vidjećemo kasnije).

Ako uslov nije ispunjen nastavlja se sa:

$$U \leftarrow d((Z)_K)$$
$$P \leftarrow p_{r(U)+1}$$

$(Z)_K = < a, <b, c >>$ je broj K-te instrukcije. Zato je $U = < b, c >$ kod instrukcije koju treba izvršiti. Varijabla koja se pominje u K-toj instrukciji biće (c+1)-va, odnosno (d(U)+1)-va, u našoj listi varijabli. Njena vrijednost se dobija kao eksponent u kojem P dijeli S, što i obavlja sljedeći programski segment:

IF l(U) = 0 GOTO N
IF l(U) = 1 GOTO A
IF $\sim(P|S)$ GOTO N
IF l(U) = 2 GOTO M

Ako je l(U) = 0, onda instrukcija koju treba izvršiti je tipa $V \leftarrow V$ pa, zapravo, ne treba ništa raditi.

Ako je l(U) = 1, onda je instrukcija je tipa $V \leftarrow V + 1$, pa tada treba dodati 1 na eksponent od P u faktorizaciji S. Izračunavanje se nastavlja sa GOTO A.

Ako je $l(U) \neq 0, 1$, tada je instrukcija koju treba izvršiti tipa $V \leftarrow V - 1$ ili IF $V \neq 0$ GOTO L. U oba slučaja, ako P nije djelioc od S, odnosno vrijednost varijable V je 0, ne treba ništa raditi. Ako $P \mid S$ i l(U) = 2 tada se izvršava GOTO M , da bi se oduzelo 1 od eksponenta kojim P dijeli S.

Nastavljamo sa:

$K \leftarrow \min_{i \leq D(Z)}[l((Z)_i) + 2 = l(U)]$
GOTO C

Ako je $l(U) > 2$ i $P \mid S$ instrukcija je oblika IF $V \neq 0$ GOTO L, gdje V ima vrijednost različitu od 0, a L je labela koja se nalazi na poziciji l(U) - 2 u listi labela. Tako, sljedeća instrukcija treba da bude prva iz liste sa tom labelom. Ako nema instrukcije sa navedenom labelom K dobija vrijednost 0, što će obezbijediti završetak programa u sljedećem ciklusu. U oba slučaja, GOTO C vraća izvršenje na početak ciklusa dekodiranja i izvršenja instrukcija.

Nastavljamo sa:

```
[M]  S ← S/P      // cjelobrojno dijeljenje
     GOTO N
[A]  S ← S*P
[N]  K ← K + 1
     GOTO C
```

Dodavanje i oduzimanje 1 od vrijednosti varijable pomenute u instrukciji postiže se množenjem i dijeljenjem S sa P, respektivno. Programski brojač se povećava za 1 i proces nastavlja sljedećom instrukcijom.

Na kraju programa treba još dodati instrukciju:

[F] $Y \leftarrow (S)_1$

jer se na kraju programa vrijednost za izlaz Y nalazi kao eksponent prostog broja p_1 (=2) u S.

Pošto smo tako u potpunosti konstruisali program U_n možemo ga prikazati u cjelini:

$Z \leftarrow X_{n+1} + 1$
$S \leftarrow \prod_{i=1}^{n} (p_{2i})^{X_i}$
$K \leftarrow 1$
[C] IF K = D(Z) + 1 $\vee$ K = 0 GOTO F
$U \leftarrow d((Z)_K)$
$P \leftarrow p_{r(U)+1}$
IF l(U) = 0 GOTO N
IF l(U) = 1 GOTO A
IF $\sim$(P|S) GOTO N
IF l(U) = 2 GOTO M
$K \leftarrow \min_{i \leq D(Z)}[l((Z)i) + 2 = l(U)]$
GOTO C
[M] $S \leftarrow S/P$
GOTO N
[A] $S \leftarrow S*P$
[N] $K \leftarrow K + 1$
GOTO C
[F] $Y \leftarrow (S)_1$

Slika 4.1 Univerzalni program U_n

Za svako n > 0, niz

$$\Phi^{(n)}(x_1,...,x_n,0),\ \Phi^{(n)}(x_1,...,x_n,1),...$$

daje redom sve parcijalno izračunljive funkcije sa n varijabli. Kada želimo da istaknemo ovaj aspekat pisaćemo:

$$\Phi_y^{(n)}(x_1,...,x_n) = \Phi^{(n)}(x_1,...,x_n,y).$$

Često je pogodno da se izostavi superskript (n) kada se radi o funkcijama sa jednom varijablom. U tim slučajevima pisaćemo:

$$\Phi_y(x) = \Phi(x,y) = \Phi^{(1)}(x).$$

Uz jednostavnu modifikaciju programa U_n možemo da konstruišemo i program za sljedeći predikat:

$STP^{(n)}(x_1,...,x_n,y,t)$ ⇔ Program sa brojem y zaustavlja rad poslije t koraka kada mu se da ulaz $x_1,...,x_n$

⇔ Postoji izračunavanje programa sa brojem y dužine ≤ t +1, sa ulazom $x_1,...,x_n$.

Dokazaćemo:

Teorema 4.2. (Teorema o broju koraka - Step Counter)

Predikat $STP^{(n)}(x_1,...,x_n,y,t)$ je izračunljiv.

Intuitivno je lako vidjeti zašto je ova teorema istinita. Potrebno je "prosto" pustiti da se program y izvršava za t koraka. Ako program zustavi rad predikat je istinit, inače je neistinit.

Dokaz.

Dokaz se i bazira na ideji da se univerzalni program modifikuje dodavanjem varijable Q koja igra ulogu kontrolnog brojača koraka. Za svaki korak ciklusa Q se uveća za 1, pa program "zna" kada je prekoračen neki unaprijed zadati broj koraka.

Program završava rad sa Y = 0 ako se program y ne zaustavlja poslije X_{n+2} (t) koraka, a sa Y = 1 ako se završava. Sa (*) su označene instrukcije koje se razlikuju od univerzalnog programa.

```
        Z ← X_{n+1} + 1
        S ← ∏_{i=1}^{n} (p_{2i})^{X_i}
        K ← 1
[C]     Q ← Q + 1                                   (*)
        IF Q > X_{n+2} + 1 GOTO E                   (*)
        IF K = D(Z) + 1 ˅ K = 0 GOTO F
        U ← d((Z)_K)
        P ← p_{r(U)+1}
        IF l(U) = 0 GOTO N
        IF l(U) = 1 GOTO A
        IF ~(P|S) GOTO N
        IF l(U) = 2 GOTO M
        K ← min_{i≤D(Z)}[ l((Z)_i) + 2 = l(U) ]
        GOTO C
[M]     S ← S/P
        GOTO N
[A]     S ← S*P
[N]     K ← K + 1
        GOTO C
[F]     Y ← 1                                       (*)
```

Slika 4.1 Program za $STP^{(n)}(X_1,...,X_n,X_{n+1},X_{n+2})$

Vježbe

1. Pokazati da za svako u postoji beskonačno mnogo različitih brojeva v takvih da je, za svako x,
$\Phi_u(x)=\Phi_v(x)$.

2. Pokazati da je $STP^{(n)}$ primitivno rekurzivna.