GLAVA 2

Programi i izračunljive funkcije

I cijeli ovi besporeci
po poretku nekome sljeduju.
Nad svom ovom grdnom mješavinom
opet umna sila toržestvuje.
Njegoš

2. Primjeri programa u jeziku S

Primjer 2.1. Posmatrajmo sljedeći program:

```
[A]   X ← X - 1
      Y ← Y + 1
      IF  X ≠ 0  GOTO A
```

Ako početna vrijednost variable X nije 0, ovaj program "kopira" vrijednost X u Y, i istovremeno smanjuje vrijednost X na 0. Ako je početna vrijednost za X jednaka 0, tada program završava rad sa vrijednošću Y jednakoj 1. Za gornji program možemo reći da izračunava funkciju:

$$f(x)=\begin{cases} 1, & ako\ je\ \ x=0 \\ x, & inace \end{cases}$$

Program završava rad prilikom izvršavanja treće instrukcije kada X postane jednako 0. Tada uslov X≠0 nije ispunjen, pa nema skoka na instrukciju sa labelom [A], te program pokušava da izvrši sljedeću instrukciju, koje nema, pa završava rad. U jeziku S program završava rad i kada treba izvršiti instrukciju sa labelom L, a takva instrukcija ne postoji. Koristićemo slovo E u IF instrukciji, kao labelu nepostojeće instrukcije.

Primjer 2.2.
Iako je prethodni primjer dobro definisan program jezika S, možemo ga posmatrati kao pokušaj da se napravi program koji kopira vrijednost X u Y, ali da je pri tome napravljena greška (bag), jer za slučaj početne vrijednosti X jednake 0, program ne daje željeni rezultat. Sljedeći program ispravlja ovaj "nedostatak":

```
[A]   IF  X ≠ 0  GOTO B
      Z ← Z + 1
      IF  Z ≠ 0  GOTO E
[B]   X ← X - 1
      Y ← Y + 1
      Z ← Z + 1
      IF  Z ≠ 0  GOTO A
```

Lako se možemo uvjeriti da ovaj program kopira vrijednost X u Y, za sve početne vrijednosti X. Zato, možemo reći da program izračunava funkciju f(x) = x.

U gornjem programu, uloga varijable Z može, na prvi pogled, izgledati nejasna. Varijabla Z je iskorišćena za "pravljenje" bezuslovnog skoka. Tako programski segment

```
Z ← Z + 1
IF  Z ≠ 0  GOTO L
```

ima efekat instrukcije tipa

```
GOTO L.
```

Sada instrukciju GOTO L možemo uključiti u naš programski jezik, znajući da je ona zapravo skraćeno zapisan programski segment, pomoću koga smo je "napravili". Ovakve skraćenice nazivaćemo makro instrukcijama jezika S.

Primjer 2.3.

Program u Primjeru 2.2 zaista kopira X u Y, ali pri tome se sadržaj X umanjuje na 0. Uobičajeno je da, u programskim jezicima, postoji instrukcija za kopiranje vrijednosti jedne varijable u drugu, ali bez uništavanja originalne vrijednosti. Zato ćemo sada program 2.2. "debagirati" tako da pri kopiranju, X ne izgubi originalni sadržaj:

```
[A]  IF  X ≠ 0  GOTO B
     GOTO C
[B]  X ← X - 1
     Y ← Y + 1
     Z ← Z + 1
     GOTO A
[C]  IF  Z ≠ 0  GOTO D
     GOTO E
[D]  Z ← Z - 1
     X ← X + 1
     GOTO C
```

U prvom ciklusu (petlji) program kopira X u Y i Z, dok se u drugom ciklusu, Z kopira nazad u X. Kada program završi rad i X i Y će imati početnu vrijednost sadržanu u varijabli X, a Z će biti jednako 0. Koristićemo ovaj program kao makro instrukciju tipa V ← V'.

Pogodno je da uvedemo i makro instrukciju tipa V ← 0, koje se dobija sljedećim programskim segmentom

```
[L]  V ← V' - 1.
     IF  V ≠ 0  GOTO L
```

Primjer 2.4. Program koji izračunava funkciju $f(x_1,x_2)=x_1+x_2$.

```
        Y ← X1
        Z ← X2
[B]     IF  Z ≠ 0  GOTO A
        GOTO E
[A]     Z ← Z - 1
        Y ← Y + 1
        GOTO B
```

Primjer 2.5. Program za izračunavanje funkcije $f(x_1,x_2)=x_1 \cdot x_2$.

```
        Z2 ← X2
[B]     IF  Z2 ≠ 0  GOTO A
        GOTO E
[A]     Z2 ← Z2 - 1
        Z1 ← X1 + Y
        Y ←  Z1
        GOTO B
```

Naravno, instrukcija $Z_1 \leftarrow X_1 + Y$ je makro proširenje, dobijeno programom 2.4., koje smo sada iskoristili radi skraćenog pisanja programa.

Primjer 2.6. Posmatrajmo sljedeći program

```
        Y ← X1
        Z ← X2
[C]     IF  Z ≠ 0  GOTO A
        GOTO E
[A]     IF  Y ≠ 0  GOTO B
        GOTO A
[B]     Y ← Y - 1
        Z ← Z - 1
        GOTO C
```

Ako program počne sa radom sa $X_1 = m$, $X_2 = n$, za $m \geq n$ program završava rad sa $Y = m - n$, odnsono izračunava funkciju $f(x_1,x_2)=x_1 - x_2$. Može se lako uočiti da, u slučaju $m < n$, program ulazi u beskonačnu petlju:

```
[A]     IF  Y ≠ 0  GOTO B
        GOTO A
```

pa nikada ne završava rad.

Zato kažemo da program izračunava parcijalnu funkciju:

$$g(x_1,x_2)=\begin{cases} x_1 - x_2 & ako\ je\ x_1 \geq x_2 \\ nedefinisano & ako\ je\ x_1 < x_2 \end{cases}$$

Vježbe

1. Napisati program u jeziku S (slobodno je korišćenje makro proširenja) kojim se izračunava funkcija f(x)= 3x.
2. Neka je f(x) = 1 ako je x paran, a f(x) = 0 ako je x neparan. Napisati program u jeziku S koji izračunava funkciju f.
3. Neka je f(x) = 1 ako je x paran, a f(x) nedefinisano ako je x neparan. Napisati program u jeziku S koji izračunava funkciju f.