

## Programi i izračunljive funkcije

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

**Njegoš**

### 1. Programski jezik S

[ta se može izračunati? Da li se baš sve može izračunati? Kako se može izračunavati? Odgovor na ova i druga pitanja, kojima se bavi **teorija izračunljivosti**, pokušaćemo da nađemo polazeći od elementarnih računskih operacija.

Definisaćemo, najprije, jedan poseban jezik, koji ćemo zvati jezik S. Zatim ćemo pokazati da, iako elementaran, jezik S može poslužiti i za najsloženija izračunavanja. Zapravo, pokazaćemo da je dovoljan za bilo koje izračunavanje koje se može izraziti nekim **algotitmom**. Drugim riječima, sve što se može izračunati (prikazati algoritmom) može se izraziti jezikom S, pa je na taj način jezik S **univerzalan programski jezik**. Ali, krenimo redom.

Počnimo najprije sa osnovnim elementima jezika S. Vrijable jezika S su slovne oznake za nenegativne cijele brojeve.

Slova  $X_1, X_2, X_3, \dots$  zvaćemo **ulaznim varijablama**.

Slova  $Z_1, Z_2, Z_3, \dots$  zvaćemo **lokalnim varijablama**.

Slovo Y zvaćemo **izlaznom varijablom**.

Ponekad ćemo, zbog jasnoće teksta, izostavljati indeks varijable, to jest umjesto  $X_1$ , pisaćemo X ili umjesto  $Z_1$  pisaćemo Z (za varijable sa indeksom 1).

U jeziku S moći ćemo da pišemo različite **instrukcije**; a pod **programom** ćemo podrazumijevati konačnu **listu instrukcija**.

Na primjer, za svaku varijablu V možemo definisati instrukciju:

$$V \leftarrow V + 1.$$

Jedan prost primjer programa u jeziku S:

$$\begin{aligned} X &\leftarrow X + 1 \\ X &\leftarrow X + 1 \end{aligned}$$

"**Izvršavanje**" ovog programa ima za efekat da se poveća vrijednost varijable X za 2.

Pored varijabli, za potrebe pisanja programa, korist ćemo i **labele** (oznake instrukcija). U tu svrhu korist ćemo slova:

$A_1, B_1, C_1, D_1, E_1, A_2, B_2, C_2, D_2, E_2, A_3, \dots$

Sljedeća tabela prikazuje instrukcije jezika S.

**Tabela 1.1** Instrukcije jezika S

<b>Instrukcija</b>	<b>Interpretacija</b>
<b><math>V \leftarrow V + 1</math></b>	Povećaj vrijednost varijable V za 1.
<b><math>V \leftarrow V - 1</math></b> jednaka nuli,  vrijednost	Ako je vrijednost varijable V  ne radi ništa, inače smanji  varijable V za 1.
<b>IF <math>V \neq 0</math> GOTO L</b> jednaka 0,  ima labelu L;  instrukcijom.	<b>Ako</b> vrijednost varijable V nije  nastavi sa instrukcijom koja  inače nastavi sa sljedećom

Gornje instrukcije zvaćemo inkrementiranje, dekrementiranje i uslovni skok, respektivno. Usvajamo, takođe da, na početku izvršavanja programa, sve lokalne varijable  $Z_i$  i izlazna varijabla Y imaju vrijednost 0, dok ulazne varijable imaju zadate početne vrijednosti.

Instrukcije mogu biti sa ili bez labele. Kada instrukcija ima labelu, labela će biti napisana unutar uglastih zagrada, kao u primjeru:

[B]  $Z \leftarrow Z - 1$

Kasnije ćemo dati jednu još formalniju definiciju jezika S, ali za sada probajmo njegovu upotrebljivost na nekim primjerima.