

□ Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

\*\*\*\*\*

Druga domaća zadaća iz predmeta  
„Uvod u teoriju računarstva“

Zadatak broj 2082

Zagreb, lipanj 2009.

## Druga domaća zadaća iz predmeta „Uvod u teoriju računarstva“

**Student:** \*\*\*\*\*

**Matični broj studenta:** \*\*\*\*\*

### Zadatak broj 2082:

Konstruirati Turingov stroj koji za ulazni aritmetički niz zapisan u postfiks notaciji gradi ekvivalentni izraz u prefiks notaciji. Turingov stroj zapisuje rezultat na neku od traka. U aritmetičkom izrazu se ne pojavljuju zagrade, a umjesto brojeva, javlja se varijabla K. Može se pretpostaviti da su ulazni nizovi uvijek ispravno zadani aritmetički izrazi.

Npr.	KK*	izgradi se izraz	*KK
	KKK*+	izgradi se izraz	+K*KK
	KK/	izgradi se izraz	/KK
	KKK*+KK/-	izgradi se izraz	-+K*KK/KK

Pokazati rad na primjeru niza KKK\*+KK/- .

## UVOD

### O zadatku

Zadatak koji sam dobio se bavi problematikom prebacivanja aritmetičkog izraza pisanog u postfiks notaciji u aritmetički izraz pisan u prefiks notaciji. Naoko, ovo je uistinu kompliciran zadatak. Naime, u praksi se većina prebacivanja kod notacija bazira na prebacivanje infiks notacije u jednu od dvije navedene, odnosno obrnuto. No, samo prebacivanje postfiks notacije u prefiks notaciju je nešto veća problematika. O razlozima toga ću govoriti kasnije kada budem nešto dublje ušao u samo pojašnjenje ovih notacija. Infiks notacija je standardna notacija, koja je ljudskom umu intuitivna i samim time znatno razumljivija. Dakle, razlog zašto je ovaj u zadatku zadani tip prebacivanja između notacija pomalo problematičan je iz prostoga razloga što, da bi ga razumjeli, moramo problemu pristupiti na drugoj razini logičkog razmišljanja.

Nadalje, ovu zadaću treba ostvariti pomoću Turingovog stroja. Turingov stroj sam po sebi je primitivan i jednostavan, te se samim time rješavanju ovog problema mora pribjeći osnovnoj logici, i detaljnom razrađivanju jednostavnih ali učinkovitih algoritama, koji uz ove dvije navedene osobine moraju biti i „fail safe“, odnosno moramo biti sigurni kako će oni svoj posao odrađivati u bilo kojem slučaju ispravno zadanoga aritmetičkoga izraza u postfiks notaciji. Tu govorimo o mnogo različitih slučajeva aritmetičkih izraza - od kratkih i jednostavnih, pa sve do dužih i kompliciranih izraza.

### Turingov stroj

Turingov stroj je izrazito jednostavan apstraktni stroj koji manipulira simbolima na jednoj ili više traka. Unatoč svojoj jednostavnosti, Turingov stroj uz pravilno osmišljavanje algoritama, odnosno prijelaza, može simulirati bilo koji računalni algoritam.

Sam stroj je osmislio Alan Turing, 1936. godine. Turingov stroj nije zamišljen kao praktično rješenje, nego kao apstraktna interpretacija. Dakle, Turingov stroj nije stroj koji bi se u stvarnosti koristio, nego se koristi kao interpretacija ograničenja mehaničkoga računala. Turingov stroj nam zapravo ukazuje na složenost algoritma koju je moguće ostvariti pomoću jednostavnog mehaničkog računala. Turingov stroj predstavlja duboki uvid u samo funkcioniranje algoritama, te visoku razinu primjene logike i matematike.

Osnovni Turingov stroj se „fizički“ sastoji od dvoje komponente: glave za čitanje/pisanje, te trake po kojoj ta glava piše, odnosno sa koje čita. Traka je podijeljena na ćelije, koje nam predstavljaju prostor za upis simbola. Svaka ćelija može primiti jedan simbol. Ova traka može biti ograničena sa obje strane (odnosno fiksnih dimenzija), neograničena na

lijevu ili desnu stranu, ili neograničena na obje strane.

Glava za čitanje/pisanje se po kreće po traci lijevo ili desno. Glavom upravlja upravljačka jedinica. Upravljačka jedinica uzima podatke koje joj glava šalje, odnosno šalje podatke koje glava treba zapisati u trenutnu ćeliju, te naposljetku ona donosi odluku o sljedećem koraku djelovanja glave. Upravljačka jedinica tijekom rada donosi odluku na temelju dva podatka:

- 1) Stanje
- 2) Znak na traci

Na temelju pročitanoog znaka i stanja jedinice, određuje se:

- 1) U koje novo stanje prelazi upravljačka jedinica
- 2) Koji znak se zapiše na traku umjesto pročitanoog znaka
- 3) U koju stranu se miče glava za čitanje i pisanje

Turingov stroj je pri ostvarivanju potrebno definirati uređenom sedmorkom:

$$ts = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

gdje je:

$Q$  - konačan skup stanje

$\Sigma \subseteq (\Gamma - \{B\})$  - konačan skup znakova trake

$\Gamma$  - konačan skup ulaznih znakova

$\delta$  - funkcija prijelaza,  $\delta : Q \times \Gamma^2 \rightarrow Q \times (\Gamma \times \{L, S, R\})$

$q_0 \in Q$  - početno stanje

$B \in \Gamma$  - znak kojim se označava prazna ćelija

$F \subseteq Q$  - skup prihvatljivih stanja

Dozvoljeno je da je funkcija prijelaza nedefinirana za pojedine argumente. Sama funkcija prijelaza znači da ukoliko Turingov stroj koji je u datom trenutku pročitao određeni znak, tada na traku piše neki drugi znak zadan u funkciji, te se glava za čitanje pomiče u stranu koja je također definirana u funkciji.

Slijedi primjer funkcije prijelaza za Turingov stroj sa jednom trakom:

$$(q_0, [A]) \rightarrow (q_1, [B], [D])$$

$q_0$  - predstavlja stanje u kojem se trenutno stroj nalazi

$A$  - predstavlja znak koji glava čita

$q_1$  - predstavlja stanje u koji stroj prelazi

$B$  - predstavlja znak koji se piše na trenutnu lokaciju

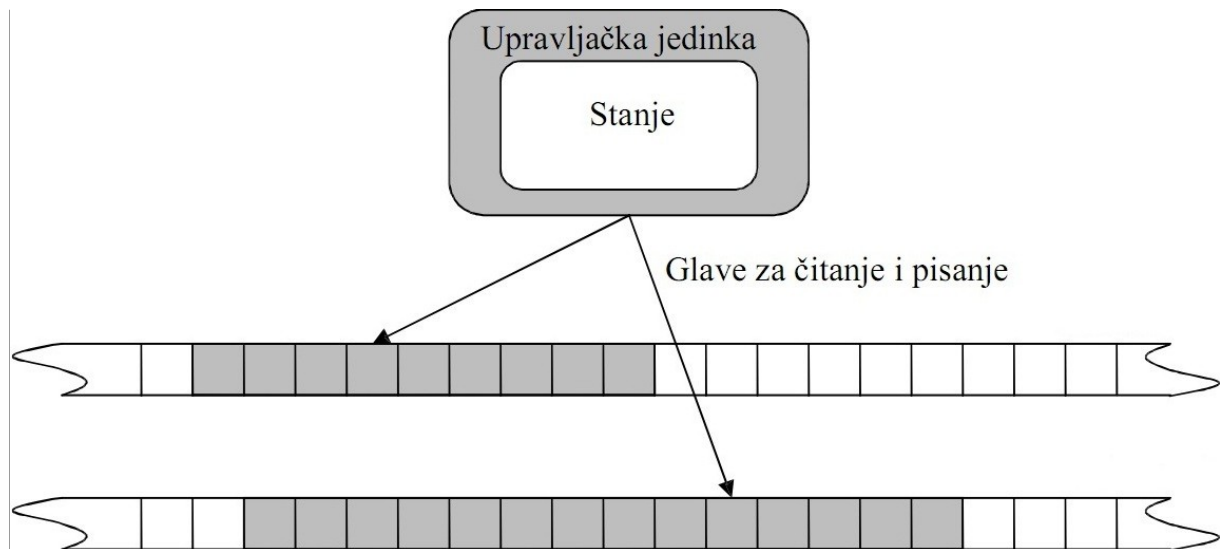
$D$  - predstavlja smjer u kojem se pomiče glava za čitanje/pisanje

Dakle, ova funkcija prijelaza nam govori sljedeće:

Ukoliko se stroj nalazi u stanju  $q_0$ , te glava čita znak  $A$ , tada stroj treba prijeći u stanje  $q_1$ , te glava zapisati umjesto znaka  $A$  novi znak  $B$ , te se pomaknuti za jednu poziciju na traci u desnu stranu.

Sve ovo što je navedeno za Turingov stroj sa jednom trakom, u pravilu vrijedi i za Turingov stroj sa više traka. Dakle, Turingov stroj sa više traka jednostavno će imati više glava za čitanje i pisanje, te ekvivalentan broj traka. Kod funkcije prijelaza će, umjesto jednoga

znaka koji će čitati, čitati dva znaka, pisati dva znaka, te će se definirati kretanje u desno ili u lijevo (ili mirovanje) za svaku od glava.



*Ilustracija: Turingov stroj sa dvije trak*

## Notacije

Notacija je način pisanja aritmetičkih izraza. Naime, svaki aritmetički izraz se sastoji od operatora i operandada. Operatori kojima se ova zadaća bavi su  $+$ ,  $-$ ,  $*$ , te  $/$ . Svaki od ovih operatora se odnosi na svoja dva operandada. Operand može biti ili eksplicitno zadan ili zadan kao drugi aritmetički izraz. U tom slučaju operand je rješenje aritmetičkog izraza.

Postoje tri tipa notacije:

**prefiksalna** – ovaj tip notacije je također poznat i kao Poljska notacija. Kod ove notacije operator dolazi prije operandada. Primjer ove notacije:

$* - 5 6 7$

**sufiksalna** – ovaj tip notacije ne također poznat i kao obrnuta Poljska notacija. Kod ove notacije operator dolazi nakon operandada. Primjer ove notacije:

$5 6 - 7 *$

**infiksalna** – ovaj tip notacije je poznat kao standardni tip notacije koji se koristi pri pisanju aritmetičkih izraza. Kod njega se operator nalazi između operatora. Primjer ove notacije:

$(5 - 6) * 7$

## OSTVARENJE

### Opći opis ostvarenja

Kao što je u zadatku i zadano, funkciju koju moj zadatak treba ostvarivati ostvario sam sa Turingovim strojem sa više traka. U mom slučaju, radi se o dvije trake, neograničene sa obje strane. Na prvoj traci se nalazi zapisan ulazni aritmetički izraz, okružen znakovima 'B', koji predstavljaju prazne ćelije. Druga traka je prazna, odnosno popunjena samo znakovima 'B'. Kako Turingov stroj prolazi kroz stanja zadana u tablici koju ću navesti kasnije, popunjava se druga traka, i to s desne na lijevu stranu. Razlog zašto je popunjavam na taj način je taj što je u prefiksna notacija u svojoj osnovi „naopaka“ postfiks notacija. Dakle, u pravilu, ono što ću raditi je sljedeće:

- pronaći ću lokaciju na traci na kojoj je početak niza
- doći ću do kraja niza
- pronaći prvi 'K', odnosno operand, s desne strane, te ga prepisati u drugu traku, a na prvoj traci na njegovo mjesto postaviti 'X'
- pronaći drugi 'K', odnosno operand, s desne strane, te ga prepisati u drugu traku, a na prvoj traci na njegovo mjesto postaviti 'X'
- nakon toga pronaći prvi operator s desne strane zadnjeg 'K' koji smo prepisali, te ga prepisati u drugu traku, a na prvoj traci na njegovo mjesto postaviti 'X'
- ponovno tražiti prvi 'K' s desne strane
- ...
- to raditi sve dok ne dođe do 'B' u potrazi za novim 'K'-om, što znači da je došao do početka niza, odnosno obradio je sve operande
- nakon toga samo popiše preostale operande redom s lijeva na desno
- kada dođe do 'B' sa desne strane, prijeđe u prihvatljivo stanje

Program završava tako što u prvoj traci na mjestu ulaznog niza ostanu samo 'X'-evi, a na drugoj traci se nalazi ulazni aritmetički niz, samo u prefiks notaciji.

## Tablica prijelaza

Kako bi prijelazi bili što lakše čitljivi i jasniji, ovdje ću ih ispisati u tablici:

q0	B, B	----->	q0	B, B	D, D	Traži početak niza
q0	+, B	----->	q1	+, B	D, D	Pronašao početak niza
q0	-, B	----->	q1	-, B	D, D	Pronašao početak niza
q0	*, B	----->	q1	*, B	D, D	Pronašao početak niza
q0	/, B	----->	q1	/, B	D, D	Pronašao početak niza
q0	K, B	----->	q1	K, B	D, D	Pronašao početak niza
q1	B, B	----->	q2	B, B	L, L	Pronašao kraj niza
q1	+, B	----->	q1	+, B	D, N	Traži kraj niza
q1	-, B	----->	q1	-, B	D, N	Traži kraj niza
q1	*, B	----->	q1	*, B	D, N	Traži kraj niza
q1	/, B	----->	q1	/, B	D, N	Traži kraj niza
q1	K, B	----->	q1	K, B	D, N	Traži kraj niza
q2	+, B	----->	q2	+, B	L, N	Traži prvi K od kraja
q2	-, B	----->	q2	-, B	L, N	Traži prvi K od kraja
q2	*, B	----->	q2	*, B	L, N	Traži prvi K od kraja
q2	/, B	----->	q2	/, B	L, N	Traži prvi K od kraja
q2	X, B	----->	q2	X, B	L, N	Traži prvi K od kraja
q2	K, B	----->	q3	X, K	L, L	Pronašao prvi K od kraja
q3	B, B	----->	q5	B, B	R, N	Došao do početka niza
q3	+, B	----->	q3	+, B	L, N	Traži drugi K od kraja
q3	-, B	----->	q3	-, B	L, N	Traži drugi K od kraja
q3	*, B	----->	q3	*, B	L, N	Traži drugi K od kraja
q3	/, B	----->	q3	/, B	L, N	Traži drugi K od kraja
q3	K, B	----->	q4	X, K	D, L	Pronašao drugi K od kraja
q4	+, B	----->	q2	X, +	L, L	Pronašao prvi op. s desne strane
q4	-, B	----->	q2	X, -	L, L	Pronašao prvi op. s desne strane
q4	*, B	----->	q2	X, *	L, L	Pronašao prvi op. s desne strane
q4	/, B	----->	q2	X, /	L, L	Pronašao prvi op. s desne strane
q4	K, B	----->	q4	K, B	R, N	Traži prvi operand s desne strane
q4	X, B	----->	q4	X, B	R, N	Traži prvi operand s desne strane
q5	B, B	----->	q6	B, B	N, N	Kraj obrade
q5	+, B	----->	q5	X, +	R, N	Našao operand s desna, traži i dalje
q5	-, B	----->	q5	X, -	R, N	Našao operand s desna, traži i dalje
q5	*, B	----->	q5	X, *	R, N	Našao operand s desna, traži i dalje
q5	/, B	----->	q5	X, /	R, N	Našao operand s desna, traži i dalje
q5	X, B	----->	q5	X, B	R, N	Traži operand s desna

Rad stroja za primjer **KKK\*+KK/-**



Prvi red tablice predstavlja ulaznu traku, drugi red položaj prve glave, u srednjem redu piše stanje u kojemu se trenutno nalazi stroj, četvrti red predstavlja izlaznu traku, a peti red predstavlja položaj druge glave.

Stroj je u q0 sve dok ne pronađe prvu ćeliju koja nije „prazna“.

B	B	K	K	K	*	+	K	K	/	-	B	B
+												
q0												

B	B	B	B	B	B	B	B	B	B	B	B	B
+												

Kada dođe do prve takve ćelije, prelazi u q1

B	B	K	K	K	*	+	K	K	/	-	B	B
+												
q1												

B	B	B	B	B	B	B	B	B	B	B	B	B
+												

Zatim pronađe kraj niza, te prijeđe u q2, te krene lijevo u potragu za prvim 'K', druga glava nakon ovoga se pomiče samo kada zapisuje

B	B	K	K	K	*	+	K	K	/	-	B	B
+												
q2												
B	B	B	B	B	B	B	B	B	B	B	B	B
+												

Kada ga nađe, zapiše ga, pa prelazi u stanje q3 te traži i drugi 'K'

B	B	K	K	K	*	+	K	X	/	-	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---

+

q3

B   B   B   B   B   B   B   B   B   B   K   B   B

+

Zatim kad i njega nađe i zapiše, ide desno sve dok ne nađe operator

B   B   K   K   K   \*   +   X   X   /   -   B   B

+

q4

B   B   B   B   B   B   B   B   B   K   K   B   B

+

Nakon što ga zapiše, ide lijevo te ponovno prolazi kroz cijelu proceduru, sve dok ne dođe do početka niza, te onda samo prepisuje operande koji su ostali s lijeva na desno.

B   B   X   X   X   X   X   X   X   X   -   B   B

+

q5

B   B   B   +   K   \*   K   K   /   K   K   B   B

+

Nakon što završi i sa time, odnosno dođe u stanju q5 do „praznih“ ćelija sa desne strane, prelazi u stanje q6, koje je prihvatljivo te završava sa obradom.

q6

No, međutim, zanimljivo je vidjeti kroz ovako jednostavan stroj kako zapravo ostvarujemo algoritme, i kako oni funkcioniraju.