

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Matej Bubnjević

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

Zadatak broj 2007

Zagreb, lipanj 2011.

Druga domaća zadaća iz predmeta „Uvod u teoriju računarstva“

Student: Matej Bubnjević

Matični broj studenta: 0036454488

Zadatak broj 2007: Definirati (nije potrebno programski ostvariti) TS (s više traka) koji će za ulazni aritmetički izraz u infiks notaciji izgraditi ekvivalentni izraz u prefiks notaciji i zapisati ga na neku od traka. U aritmetičkom izrazu se ne pojavljuju zagrade, a umjesto brojeva se javlja varijabla K. Može se pretpostaviti da su ulazni nizovi uvijek ispravno zadani aritmetički izrazi.

Npr.	$K * K$	izgradi se izraz	$* K K$
	$K + K * K$	izgradi se izraz	$+ K * K K$
	K / K	izgradi se izraz	$/ K K$
	$K + K * K - K / K$	izgradi se izraz	$+ K - * K K / K K$

Pokazati rad na primjeru niza $K + K * K - K / K$

UVOD

Dobiveni zadatak se bavi definiranjem Turingovog stroja s više traka koji za aritmetički niz u infiksnoj notaciji zapisan na ulaznoj traci stvori aritmetički u prefiksnoj notaciji zapisujući ga na radnu traku. Model Turingovog stroja koji će riješiti ovaj problem će dakle biti Turingov stroj s k dvostrano beskonačnih traka i k glava za čitanje i pisanje. Prvo će biti objašnjen osnovni model Turingovog stroja pa njegov proširen model i zatim notacije. Infiksna notacija je uobičajena notacija aritmetičkih i logičkih formula koja je ljudskom umu intuitivna i samim time najrazumljivija. Računala je ne mogu tako lako parsirati kao prefiksnu notaciju ili postfiksnu notaciju, ali je mnogi programski jezici koriste kako bi iskoristili pogodnost njene sveopće upoznatosti. U infiksnoj notaciji, za razliku od prefiksni i postfiksni notacija, zagrade koje okružuju skupine operanada i operatora su nužne kako bi se naznačio redoslijed u kojem bi se operacije trebale obaviti. U odsustvu zagrada, određena pravila prednosti i asocijativnosti određuju redoslijed operacija. Prefiksnu notaciju ili prefiksni sustav oznaka je izmislio poljski logičar Jan Lukasiewicz oko 1920. kako bi pojednostavnio sustav oznaka propozicijskog računa. Ponekad se ova notacija naziva i poljska notacija kako bi se naglasila njena nacionalnost. Iako se više ne koristi toliko učestalo u logici, našla je uporabu u računarstvu. Prefiksna notacija ima široku primjenu u Lispovim S-izrazima, gdje su zagrade potrebne zbog aritmetičkih operatora promjenjive mjesnosti (ariteta). Srodna postfiksna notacija (ili „obrnuta poljska notacija“) se često koristi u stogovno orijentiranim programskim jezicima, te predstavlja načelo djelovanja određenih kalkulatora, posebice onih od tvrtke Hewlett-Packard.

Osnovni model Turingovog stroja

Turingovi strojevi su iznimno jednostavni apstraktni uređaji za manipulaciju znakovima (simbolima) koji unatoč jednostavnosti dizajna mogu biti prilagođeni da simuliraju logiku bilo kojeg računalnog algoritma. Opisao ih je 1936. Alan Turing. Turingovi strojevi ne koriste se u praktične svrhe, već u misaonim eksperimentima, gdje najvažniju primjenu nalaze u istraživanju granica mogućnosti izračunavanja računalnim algoritmima. Proučavanje njihovih svojstava pruža dalekosežne uvide u pitanja računarske znanosti i teorije složenosti.

Na slici 1 prikazan je osnovni model Turingovog stroja. Osnovni dijelovi Turingovog stroja su upravljačka jedinka, glava za čitanje i pisanje te ulazna traka. Upravljačka jedinka jest u jednom od konačnog broja stanja. Turingov stroj nakon čitanja znaka ulazne trake zapiše novi znak na traku umjesto pročitano znaka. Nadalje, glava za čitanje i pisanje miče se u lijevo i desno. Traka ima krajnje lijevu ćeliju, dok je beskonačna na desnu stranu. Na početku rada n krajnje lijevih ćelija sadrže niz w , gdje je $|w|=n$ i $n \geq 0$. Na ostatku trake su prazne ćelije koje se označavaju znakom B . Ulazni znakovi niza w i znakovi koje TS zapisuje na ulaznu traku čine skup znakova trake.



Slika 1 Model Turingovog stroja

Tijekom rada upravljačka jedinka donosi odluku na temelju dva podatka:

- 1) stanje;
- 2) znak na traci

Na temelju pročitanoog znaka i stanja jedinke, Turingov stroj odlučuje:

- 1) u koje novo stanje prelazi upravljačka jedinka;
- 2) koji znak se zapiše na traku umjesto pročitanoog znaka;
- 3) u koju stranu se miče glava za čitanje i pisanje.

Turingov stroj (TS) formalno se zadaje uređenom sedmorkom:

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

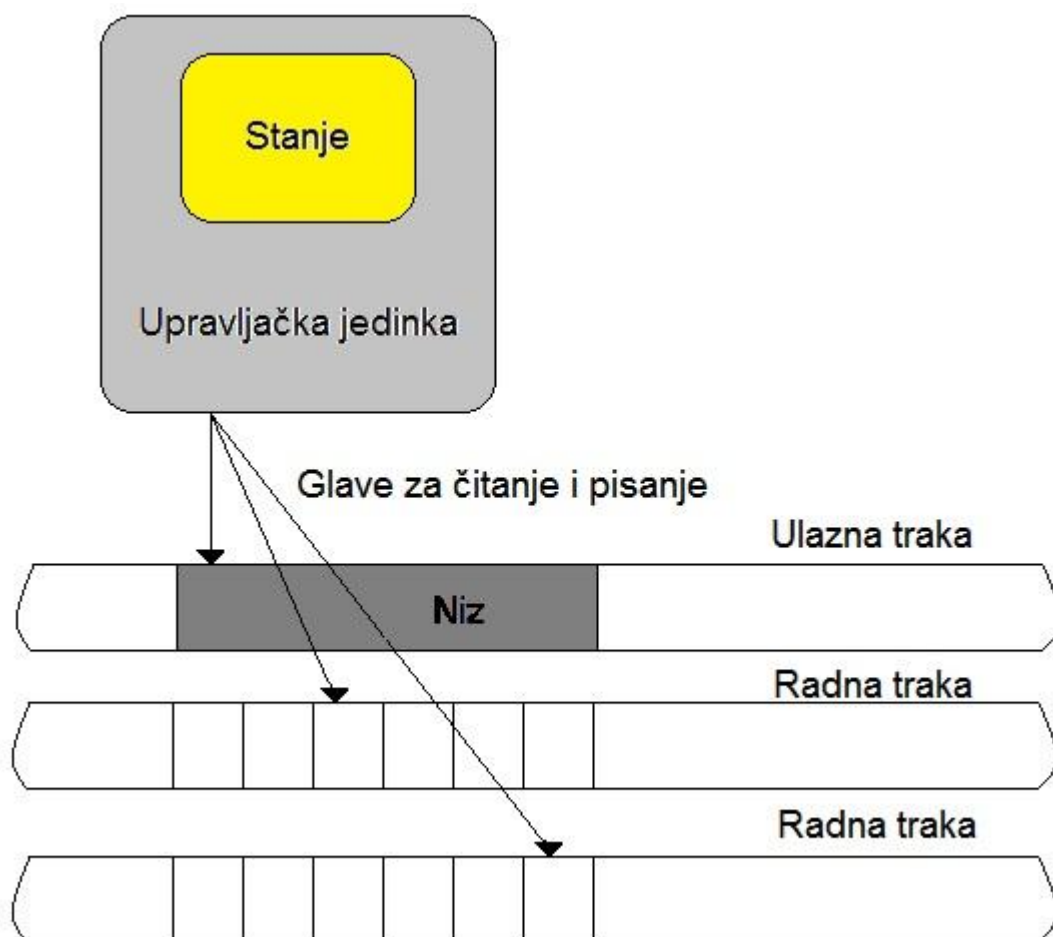
gdje je:

- Q - konačan skup stanja;
- Γ - konačan skup znakova trake;
- $B \in \Gamma$ - znak kojim se označava prazna ćelija;
- $\Sigma \subseteq (\Gamma - \{B\})$ - konačan skup ulaznih znakova;

δ - funkcija prijelaza $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$, gdje L i R označavaju pomak glave u lijevo i desno
 $q_0 \in F$ - početno stanje;
 $F \subseteq Q$ - skup prihvatljivih stanja.

Dozvoljava se da je funkcija prijelaza δ nedefinirana za pojedine argumente. Funkcija prijelaza $\delta(q, V)=(p, Z, W)$ određuje da TS iz stanja q ($q \in Q$) čitanjem znaka V ($V \in \Gamma$) prelazi u stanje p , na traku zapiše znak Z ($Z \in \Gamma$) umjesto znaka V , a glava za čitanje i pisanje miče se u lijevo ili desno ovisno o W ($W \in \{L, R\}$).

Prošireni model Turingovog stroja – TS s višestrukim trakama



Slika 2 Model TS s višestrukim trakama

Model TS s višestrukim dvostranim beskonačnim trakama prikazan je na slici 2. TS ima k glava za čitanje i pisanje i k dvostrano beskonačnih traka. Upravljačka jedinica TS donosi odluku na temelju dviju grupa parametara:

- a) stanje upravljačke jedinice;
- b) k pročitanih znakova sa k traka.

Jednim prijelazom TS:

- 1) promijeni stanje;
- 2) zapiše k znakova na k traka;
- 3) pomakne bilo koju od k glava nezavisno u desno ili lijevo.

Na jednu traku, koja se naziva ulazna traka, zapiše se niz koji se ispituje. Sve ostale trake nazivaju se radnim trakama.

TS s višestrukim trakama istovjetan je osnovnom modelu TS.

Notacije

Notacija je način pisanja aritmetičkih izraza. Naime, svaki aritmetički izraz sastoji se od operanda i operatora. Operatori mogu biti npr. $+$, $-$, $*$ ili $/$. Jedan takav operator djeluje na dva operanda koji čine jedan izraz koji može biti unutar složenog izraza te se takvi osnovni izrazi od jednog operatora i dva operanda evaluacijom zamjenjuju jednim operandom.

S obzirom na stil zapisivanja operatora i operandada postoje tri vrste notacija:

- a) Prefiksna notacija – njena osobina jest ta da su operatori postavljeni lijevo od operandada (npr. $+3\ 4$)
- b) Infiksna notacija – u njoj su operatori napisani u infiksnom stilu između operandada nad kojima djeluju (npr. $3+4$)
- c) Postfiksna notacija – notacija u kojoj svaki operator slijedi nakon svih svojih operandada (npr. $3\ 4+$)

OSTVARENJE

IDEJA OSTVARENJA

Zadatak je riješen upotrebom Turingovog stroja s tri trake, beskonačne s obje strane. Na prvu traku (ulaznu) je zapisan ulazni niz u infiksnoj notaciji, a na treću traku će biti zapisan konačni niz u prefiksnoj notaciji. Na drugu traku će se prepisivati niz do operatora + pa će se izbacivati operatori - i zatim će se izbacivati operatori * i /. Nakon toga se vraćamo na prvu traku i opet idemo do idućeg operatora + ili dok ne stignemo do kraja niza. U početku su druga i treća traka prazna odnosno njihove ćelije su ispunjene znakovima B koji označavaju praznu ćeliju. Slično će biti na prvoj traci: s lijeve i desne strane niza će u beskonačnost ići znakovi B.

Uz uvjete da znamo da je broj operanada jednak broju operatora plus jedan i da je stoga prvi znak uvijek operand K, algoritam glasi:

- 1) Prepisivanje niza s prve trake na drugu traku dok se ne stigne do operatora + ili kraja niza (ovaj dio predstavlja vanjsku petlju i stoga jedno stanje)
- 2) Kada se stigne do operatora +, zapiše se na treću traku
- 3) U drugom nizu se sada vraćamo na početak niza, odnosno do prvog znaka B, pritom zamijenjujući operatore - sa znakom Y i zapisujući operatore - na treću traku
- 4) Sada prolazimo znakovima do prvog znaka Y pritom zapisujući operatore * i / na treću traku i mijenjajući ih znakom X na njihovoj (druvoj) traci
- 5) Opet se vraćamo na početak niza na drugoj traci odnosno do znaka B te opet krećemo na desno i ovog puta zapisujemo znakove K na treću traku pritom zamijenjujući znakove K i X znakom B sve dok ne dođemo do znaka Y pa i njega zamijenim znakom B
- 6) Tada se pomaknemo u desno i opet se vraćamo na 5. korak sve dok ne stignemo do znaka B umjesto Y
- 7) Nakon što stignemo do znaka B u drugoj traci, vraćamo se opet na 1. korak
- 8) Pretvorba niza završava kada se u 2. koraku stigne do znaka B i riješi dotadašnji niz s druge trake; tada Turingov stroj prelazi u prihvatljivo stanje

Uz to, naše funkcije prijelaza će izgledati ovako:

$$\delta(q, \{V1, V2, V3\}) \rightarrow (p, \{Z1, Z2, Z3\}, \{W1, W2, W3\})$$

Ovisno o stanju q i pročitanim znakovima V1, V2 i V3 TS će prijeći u stanje p, zamijeniti znakove V1, V2 i V3 redom znakovima Z1, Z2 i Z3 te će se pripadne glave za čitanje i pisanje W1, W2 i W3 pomaknuti u desno, lijevo ili nikud (R, L, N slijedno).

DEFINICIJA OSTVARENOSTI TURINGOVOG STROJA

Sada možemo napisati definiciju našeg Turingovog stroja:

TS $M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}, \{K, +, -, *, /\}, \{K, +, -, *, /, X, Y, B\}, \delta, q_0, B, \{q_9\})$

uz sljedeće prijelaze:

1. $\delta(q_0, \{K, B, B\}) \rightarrow (q_0, \{K, K, B\}, \{R, R, N\})$
2. $\delta(q_0, \{-, B, B\}) \rightarrow (q_0, \{-, -, B\}, \{R, R, N\})$
3. $\delta(q_0, \{/, B, B\}) \rightarrow (q_0, \{/, /, B\}, \{R, R, N\})$
4. $\delta(q_0, \{*, B, B\}) \rightarrow (q_0, \{*, *, B\}, \{R, R, N\})$
5. $\delta(q_0, \{+, B, B\}) \rightarrow (q_1, \{+, B, +\}, \{N, L, R\})$
6. $\delta(q_0, \{B, B, B\}) \rightarrow (q_2, \{B, B, B\}, \{N, L, N\})$

7. $\delta(q_1, \{+, K, B\}) \rightarrow (q_1, \{+, K, B\}, \{N, L, N\})$
8. $\delta(q_1, \{+, -, B\}) \rightarrow (q_1, \{+, Y, -\}, \{N, L, R\})$
9. $\delta(q_1, \{+, /, B\}) \rightarrow (q_1, \{+, /, B\}, \{N, L, N\})$
10. $\delta(q_1, \{+, *, B\}) \rightarrow (q_1, \{+, *, B\}, \{N, L, N\})$
11. $\delta(q_1, \{+, B, B\}) \rightarrow (q_3, \{+, B, B\}, \{N, R, N\})$

12. $\delta(q_3, \{+, K, B\}) \rightarrow (q_3, \{+, K, B\}, \{N, R, N\})$
13. $\delta(q_3, \{+, *, B\}) \rightarrow (q_3, \{+, X, *\}, \{N, R, R\})$
14. $\delta(q_3, \{+, /, B\}) \rightarrow (q_3, \{+, X, /\}, \{N, R, R\})$
15. $\delta(q_3, \{+, Y, B\}) \rightarrow (q_4, \{+, Y, B\}, \{N, L, N\})$
16. $\delta(q_3, \{+, B, B\}) \rightarrow (q_4, \{+, B, B\}, \{N, L, N\})$

17. $\delta(q_4, \{+, K, B\}) \rightarrow (q_4, \{+, K, B\}, \{N, L, N\})$
18. $\delta(q_4, \{+, X, B\}) \rightarrow (q_4, \{+, X, B\}, \{N, L, N\})$
19. $\delta(q_4, \{+, B, B\}) \rightarrow (q_5, \{+, B, B\}, \{N, R, N\})$

20. $\delta(q_5, \{+, K, B\}) \rightarrow (q_5, \{+, B, K\}, \{N, R, R\})$
21. $\delta(q_5, \{+, X, B\}) \rightarrow (q_5, \{+, B, B\}, \{N, R, N\})$
22. $\delta(q_5, \{+, Y, B\}) \rightarrow (q_3, \{+, B, B\}, \{N, R, N\})$
23. $\delta(q_5, \{+, B, B\}) \rightarrow (q_0, \{+, B, B\}, \{R, N, N\})$

24. $\delta(q_2, \{B, K, B\}) \rightarrow (q_2, \{B, K, B\}, \{N, L, N\})$
 25. $\delta(q_2, \{B, -, B\}) \rightarrow (q_2, \{B, Y, -\}, \{N, L, R\})$
 26. $\delta(q_2, \{B, /, B\}) \rightarrow (q_2, \{B, /, B\}, \{N, L, N\})$
 27. $\delta(q_2, \{B, *, B\}) \rightarrow (q_2, \{B, *, B\}, \{N, L, N\})$
 28. $\delta(q_2, \{B, B, B\}) \rightarrow (q_6, \{B, B, B\}, \{N, R, N\})$

 29. $\delta(q_6, \{B, K, B\}) \rightarrow (q_6, \{B, K, B\}, \{N, R, N\})$
 30. $\delta(q_6, \{B, *, B\}) \rightarrow (q_6, \{B, X, *\}, \{N, R, R\})$
 31. $\delta(q_6, \{B, /, B\}) \rightarrow (q_6, \{B, X, /\}, \{N, R, R\})$
 32. $\delta(q_6, \{B, Y, B\}) \rightarrow (q_7, \{B, Y, B\}, \{N, L, N\})$
 33. $\delta(q_6, \{B, B, B\}) \rightarrow (q_7, \{B, B, B\}, \{N, L, N\})$

 34. $\delta(q_7, \{B, K, B\}) \rightarrow (q_7, \{B, K, B\}, \{N, L, N\})$
 35. $\delta(q_7, \{B, X, B\}) \rightarrow (q_7, \{B, X, B\}, \{N, L, N\})$
 36. $\delta(q_7, \{B, B, B\}) \rightarrow (q_8, \{B, B, B\}, \{N, R, N\})$

 37. $\delta(q_8, \{B, K, B\}) \rightarrow (q_8, \{B, B, K\}, \{N, R, R\})$
 38. $\delta(q_8, \{B, X, B\}) \rightarrow (q_8, \{B, B, B\}, \{N, R, N\})$
 39. $\delta(q_8, \{B, Y, B\}) \rightarrow (q_6, \{B, B, B\}, \{N, R, N\})$
 40. $\delta(q_8, \{B, B, B\}) \rightarrow (q_9, \{B, B, B\}, \{L, N, L\})$

Od 1. do 6. prijelaza prepisujemo niz na drugu traku dok ne stignemo do znaka + ili kraja niza.

Od 7. do 11. prijelaza se na drugoj traci vraćamo na početak niza pritom zamijenjujući minuse sa Y i zapisujući minuse na treću traku.

Od 12. do 16. prijelaza prolazimo niz na drugoj traci do znaka Y ili B pritom zamijenjujući * i / sa znakom X i zapisujući znakove * i / na treću traku.

Od 17. do 19. prijelaza se opet vraćamo na početak niza druge trake.

Od 20. do 23. prijelaza prolazimo drugom trakom zapisujući znakove K na treću traku i mijenjajući znakove K i X znakom B dok ne stignemo do znaka Y pa i njega zamijenimo znakom B i pomaknemo se u desno i vratimo u prethodnu petlju, a iz te petlje ispadamo ako stignemo do znaka B

Od 24. do 40. prijelaza je sve to isto samo za zadnji skup operatora i operandi i nakon toga završava izgradnja prefiks notacije zadanog niza i TS prelazi u prihvatljivo stanje.

RAD TURINGOVOG STROJA ZA PRIMJER $K + K * K - K / K$

Početno stanje Turingovog stroja je q_0 . Prva glava za čitanje i pisanje se nalazi nad prvim znakom ulaznog niza, a druga i treća negdje na traci (nebitno gdje jer su trake obostrano beskonačno) nad znakom B. Znak nad kojim je glava će biti podebljan. Trake će biti označene rednim brojevima: prva traka je ulazna, a druga i treća su radne te će ispod biti napisano trenutno stanje

Početno stanje:

1.B**K**+K*K-K/K

2.BBBBBBBB

3.BBBBBBBB

q_0

Primjenom 1. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.BBBBBBBB

q_0

Primjenom 5. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q_1

Primjenom 7. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q_1

Primjenom 11. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q_3

Primjenom 12. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q₃

Primjenom 16. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q₄

Primjenom 17. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q₄

Primjenom 19. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BKBBBBBB

3.B+BBBBBB

q₅

Primjenom 20. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BBBBBBBB

3.B+KBBBBB

q₅

Primjenom 23. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BBBBBBBB

3.B+KBBBBB

q₀

Primjenom 1. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BBKBBBBB

3.B+KBBBBB

q₀

Primjenom 4. funkcije prijelaza nastaje stanje:

1.BK+K***K**-K/K

2.BK***B**BBBB

3.B+K**B**BBBBB

q₀

Primjenom 1. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BK*K**B**BBB

3.B+K**B**BBBBB

q₀

Primjenom 2. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/**K**

2.BK*K-**B**BB

3.B+K**B**BBBBB

q₀

Primjenom 1. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K

2.BK*K-K**B**B

3.B+K**B**BBBBB

q₀

Primjenom 3. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/**K**

2.BK*K-K/**B**

3.B+K**B**BBBBB

q₀

Primjenom 1. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK*K-K/K**B**

3.B+K**B**BBBBB

q₀

Primjenom 6. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK*K-K/K**B**

3.B+K**B**BBBBB

q₂

Primjenom 24. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK*K-K/KB

3.B+K**B**BBBBB

Q₂

Primjenom 26. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK*K-**K**/KB

3.B+K**B**BBBBB

Q₂

Primjenom 24. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK*K-K/KB

3.B+K**B**BBBBB

Q₂

Primjenom 25. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK***K**YK/KB

3.B+K-**B**BBBB

Q₂

Primjenom 24. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK***K**YK/KB

3.B+K-**B**BBBB

Q₂

Primjenom 27. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**K***K**YK/KB

3.B+K-**B**BBBB

Q₂

Primjenom 24. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**K***K**YK/KB

3.B+K-**B**BBBB

q₂

Primjenom 28. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.B**K***KYK/KB

3.B+K-**B**BBBB

q₆

Primjenom 29. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK*KYK/KB

3.B+K-**B**BBBB

q₆

Primjenom 30. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK**X**KYK/KB

3.B+K-*****BBBB

q₆

Primjenom 29. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK**X**KYK/KB

3.B+K-*****BBBB

q₆

Primjenom 32. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK**X**KYK/KB

3.B+K-*****BBBB

q₇

Primjenom 34. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BK**X**KYK/KB

3.B+K-*****BBBB

q₇

Primjenom 35. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.B**K****X**KYK/KB

3.B+K-***BBBB**

Q₇

Primjenom 34. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**KXKYK/KB

3.B+K-***BBBB**

Q₇

Primjenom 36. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**KXKYK/KB

3.B+K-***BBBB**

Q₈

Primjenom 37. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**BXKYK/KB

3.B+K-***K**BBB

Q₈

Primjenom 38. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**BBKYK/KB

3.B+K-***K**BBB

Q₈

Primjenom 37. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**BBBYK/KB

3.B+K-***K**K**B**B

Q₈

Primjenom 39. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.**B**BBBBBK/KB

3.B+K-***K**K**B**B

Q₆

Primjenom 29. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK/KB

3.B+K-*KK**BB**

q₆

Primjenom 31. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK**X**KB

3.B+K-*KK/**B**

q₆

Primjenom 29. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK**X**KB

3.B+K-*KK/**B**

q₆

Primjenom 33. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK**X**KB

3.B+K-*KK/**B**

q₇

Primjenom 34. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK**X**KB

3.B+K-*KK/**B**

q₇

Primjenom 35. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK**X**KB

3.B+K-*KK/**B**

q₇

Primjenom 34. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBK**X**KB

3.B+K-*KK/**B**

q₇

Primjenom 34. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBB**KXKB**

3.B+K-*KK/**B**

q₈

Primjenom 37. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBBB**XKB**

3.B+K-*KK/**B**

q₈

Primjenom 38. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBBBB**BK**

3.B+K-*KK/**B**

q₈

Primjenom 37. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBBBB**B**

3.B+K-*KK/**B**

q₈

Primjenom 40. funkcije prijelaza nastaje stanje:

1.BK+K*K-K/K**B**

2.BBBBBBBB**B**

3.B+K-*KK/**B**

q₉

I konačno smo stigli do prihvatljivog stanja. Ne postoje daljnji prijelazi te automat staje i dobili smo niz u prefiksnoj notaciji na trećoj traci.

ZAKLJUČAK

Zadatak pretvaranja aritmetičkog izraza u infiksnoj notaciji u aritmetički izraz u prefiksnoj notaciji je riješen pomoću Turingovog stroja s tri obostrano beskonačne trake koji ima 10 stanja i 40 funkcija prijelaza te su bili korišteni pomoćni znakovi trake X i Y. Ovakav TS radi za sve aritmetičke izraze u kojima je broj operanada jednak broju operatora plus jedan, odnosno, za sve izraze u kojima je prvi član pozitivan (jer je to uvjet za prefiksnu notaciju). Prvi i najveći problem je bio osmišljavanje algoritma pretvorbe aritmetičkog izraza infiksne notacije u aritmetički izraz prefiksne notacije koji bi bio uporabiv za naš Turingov stroj. Dakako, prvo se trebalo dobro proučiti koji je odnos infiksne i prefiksne notacije i raditi sa mnogim primjerima da bi se svladalo pretvaranje. Tada je krenula potraga za algoritmom koji bi bio ispravan za sve moguće primjere aritmetičkih izraza. Npr. prva verzija algoritma je radila samo za sve pozitivne operande pa je nastao problem zbog operatora minus. Iduća verzija je mogla prihvatiti oduzimanje, ali samo ako je negativni član bio zadnji u cijelom izrazu. Onda se algoritam još dorađivao za slučajeve kada je bilo više operatora -, i konačno, kada je bilo više operatora - odvojenih operatorom +. Sada kada je bio dobiven potpuni algoritam, trebalo je definirati TS i pritom napisati funkcije prijelaza.

Ovaj bi problem npr. bio mnogo lakše riješen u programskom jeziku tipa C; zato vidimo zašto se ovakvi strojevi ne konstruiraju u praksi. Međutim, bilo je zanimljivo vidjeti kako se može smisliti algoritam za pretvaranje infiksne u prefiksnu notaciju i implementirati u Turingov stroj koji bi radio.

LITERATURA

1. Jezični procesori 1 – Siniša Srbljić
2. Wikipedia http://hr.wikipedia.org/wiki/Prefiksna_notacija
3. Wikipedia http://hr.wikipedia.org/wiki/Infiksna_notacija
4. Wikipedia http://hr.wikipedia.org/wiki/Turingov_stroj