

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Ivan Krešo

Druga domaća zadaća iz predmeta “Uvod u teoriju računarstva”

Zadatak broj 3071

Zagreb, lipanj 2010.

Druga domaća zadaća iz predmeta “Uvod u teoriju računarstva”

Student: Ivan Krešo

Matični broj studenta: 0036444890

Zadatak broj 3071: Konstruirati TS koji će za IP adresu zadanu u obliku xxx.xxx.xxx.xxx/N, gdje je N broj bitova mrežne maske, odrediti IP adresu podmreže. Pretpostaviti da je IP adresa ispravno zadana. IP adresa podmreže određuje se na sljedeći način:

1. ulazna IP adresa pretvori se u binarni oblik (32 bita)
2. mrežna maska napiše se kao 32 bitni binarni broj koji ima N vodećih jedinica i ostalo se nadopuni s nulama
3. napravi se logička I operacija nad prethodno definiranim brojevima, rezultat je IP adresa podmreže zapisana u binarnom brojevnom sustavu
4. binarna vrijednost IP adrese podmreže pretvori se u . notaciju

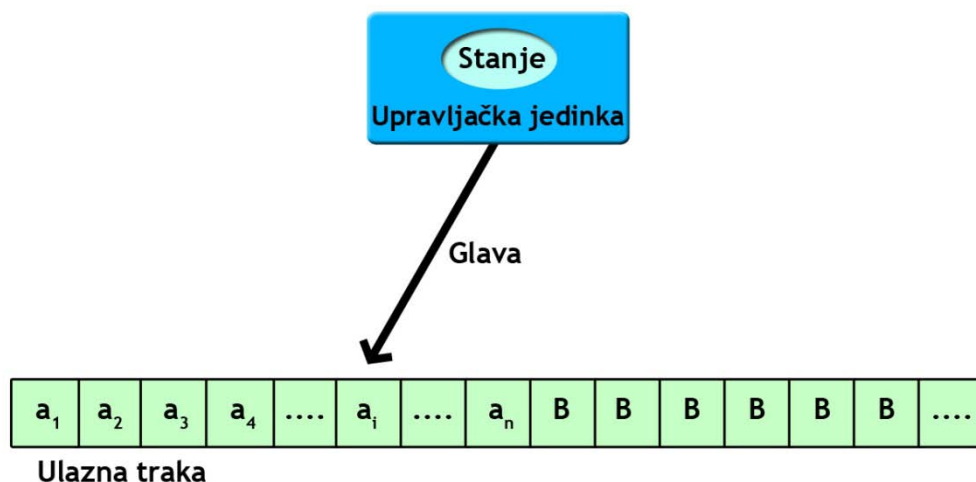
Primjer:

Ulazni niz:	136.15.192.37/17
IP adresa u binarnom zapisu:	10001000 00001111 11000000 00100101
Mrežna maska u binarnom zapisu:	11111111 11111111 10000000 00000000
IP adresa podmreže:	10001000 00001111 10000000 00000000
IP adresa podmreže u . notaciji:	136.15.128.0

Uvod

Turingov stroj predstavlja najopćenitiji matematički model računanja. Iako ga krasi jednostavnost, istih je mogućnosti računanja kao bilo koje digitalno računalo. Opisao ga je 1936. godine 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. Osnovna primjena ovog stroja jest prihvaćanje jezika. Budući da je omogućeno pisanje po ulaznoj traci, Turingov stroj koristi se za generiranje jezika i računanje cjelobrojnih funkcija.

Osnovni model Turingovog stroja čine upravljačka jedinka, ulazna traka i glava za čitanje i pisanje. Upravljačka jedinka se u svakom trenutku nalazi u jednom od konačnog broja stanja. Konačan skup stanja dijeli se na prihvatljiva i neprihvatljiva stanja. Ulazna traka sastoji se od niza ćelija u koje se spremaju znakovi trake. Traka je beskonačna na desnu stranu dok je na lijevoj strani ograničena krajnjom ćelijom. Nakon čitanja znaka sa ulazne trake, Turingov stroj zapiše novi znak na traku te se glava pomakne lijevo ili desno. Novi zapisani znak može biti jednak ili različit pročitanoj znaku. Ako glava dođe do krajnje lijeve ćelije tada se može micati samo desno. Na početku rada n krajnje lijevih ćelija sadrže niz w , gdje je $|w| = n$, $n \geq 0$. Ostale ćelije sadrže znak B koji označava praznu ćeliju. Na slici 1. prikazan je osnovni model Turingovog stroja.



Slika 1. Osnovni model Turingovog stroja

Tijekom rada upravljačka jedinka donosi odluku na temelju trenutnog stanja u kojem se nalazi i pročitano znaka. Mora odlučiti u koje novo stanje će preći, koji znak će se zapisati na traku umjesto pročitano znaka te na koju stranu će pomaknuti glavu nakon zapisivanja novog znaka. Ove odluke definirano zapisivanjem funkcija prijelaza.

Turingov stroj (TS) formalno se definira 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 u desno;
$q_0 \in Q$	-početno stanje;
$F \subseteq Q$	-skup prihvatljivih stanja.

Dozvoljeno je da funkcija prijelaza δ bude nedefinirana za pojedine argumente. Ako za neko stanje i ulazni znak nema definirane funkcije prijelaza, TS će se zaustaviti. Ako je stanje u kojem se TS nalazi kada se zaustavio prihvatljivo ulazni niz se prihvaća, a ako je stanje neprihvatljivo onda se ulazni niz ne prihvaća. Funkcija prijelaza je oblika $\delta(q, V) = (p, Z, W)$ i određuje da stroj iz stanja q ($q \in Q$) za pročitani znak V ($V \in \Gamma$) prelazi u stanje p ($p \in Q$), na mjestu znaka V zapiše znak Z ($Z \in \Gamma$) te se pomakne lijevo ili desno ovisno o W ($W \in \{L, R\}$).

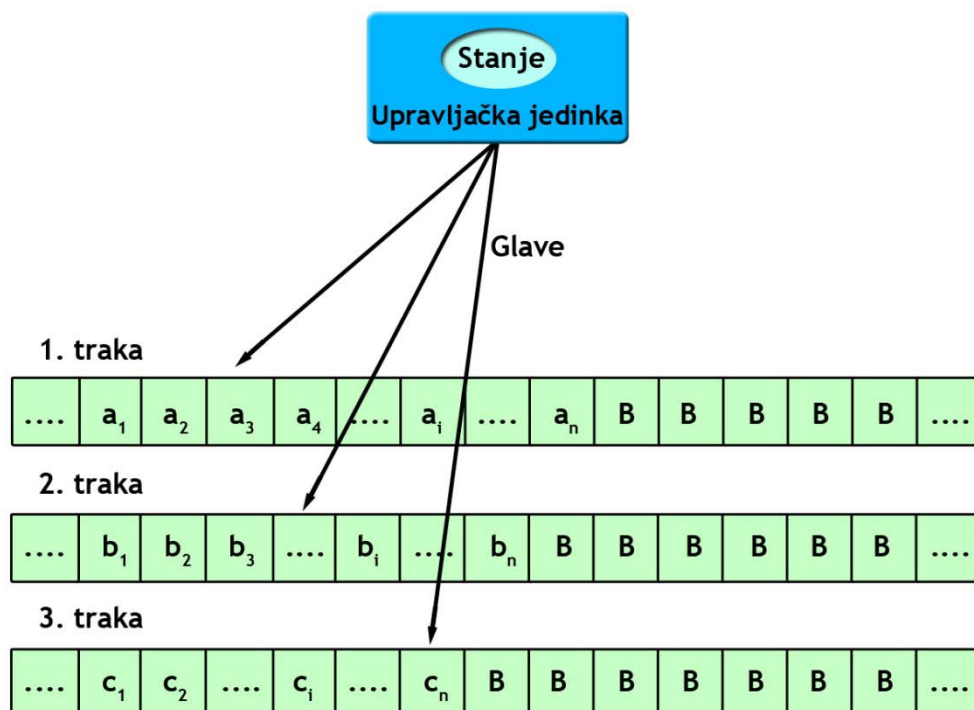
Turingovi strojevi prihvaćaju klasu rekurzivno prebrojivih jezika što povlači tvrdnju da je za bilo koji rekurzivno prebrojivi jezik moguće izgraditi TS koji generira sve nizove tog jezika. Kontekstno neovisni jezici su pravi podskup skupa rekurzivno prebrojivih jezika. Postoje rekurzivno prebrojivi jezici za koje nije moguće izgraditi TS koji će uvijek stati za bilo koji ulazni niz. Klasa jezika za koju je moguće izgraditi TS koji će uvijek stati za svaki ulazni niz, naziva se klasa

rekurzivnih jezika. Klasa rekurzivnih jezika je pravi podskup klase rekurzivno prebrojivih jezika.

Turingov stroj koristi se i za računanje vrijednosti cjelobrojnih funkcija. Funkcije koje je moguće izračunati Turingovim strojem nazivaju se parcijalno rekurzivne funkcije.

Osnovni TS model možemo proširiti na različite načine. Šest je osnovnih načina proširenja: TS sa višestrukim trakama, nedeterministički TS, TS sa višedimenzionalnim ulaznim poljem, TS sa više glava za čitanje i pisanje te neizravni TS. Svi prošireni modeli su istovjetni sa osnovnim TS modelom što znači da prihvaćaju iste jezike. Mnogi problemi jednostavnije se rješavaju pomoću proširenih TS modela.

Za potrebe rješavanja zadanog problema odabran je prošireni TS sa višestrukim obostrano beskonačnim trakama. Tri ulazne trake odabrane su kao najoptimalniji izbor. Na slici 2. prikazan je model odabranog Turingovog stroja.



Slika 2. Model višetračnog Turingovog stroja

Ostvarenje

U tekstu zadatka vidi se da je cijeli postupak određivanja adrese pod mreže sastavljen od 4 koraka. Ostvareni TS će obaviti posao kada prođe kroz spomenuta 4 koraka zadanim redoslijedom. Najprije je potrebno zadanu IP adresu pretvoriti u binarni oblik, a zatim zapisati mrežnu masku u binarnom obliku da bi se na kraju iz ova dva zapisa logičkom operacijom I dobila adresa pod mreže. Budući da je riječ o 3 zapisa koja u jednom trenutku izvođenja moraju postojati, kao najoptimalniji izbor odabran je TS sa tri trake. Sve trake će biti obostrano beskonačne jer će se stroj često puta morati vraćati na početak zapisa, a beskonačna lijeva strana olakšava taj postupak jer nije potrebna posebna oznaka početka kao prvi znak zapisa. Glave svih traka se mogu kretati lijevo i desno, ali mogu i ostati na mjestu.

Ideja ostvarenja je da se proces konstruiranja najprije podijeli u 4 faze opisane u samom zadatku. Pri prelasku na iduću fazu stroj će uvijek nastavljati tamo gdje je stala prethodna faza. U daljem tekstu objašnjena je formalna definicija te funkcije prijelaza.

Formalna definicija Turingovog stroja za ovaj zadatak:

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

Q - konačan skup stanja = $\{q_0, q_1, q_2, \dots, q_{10}, q_{11}, \dots, q_{24}\}$ - ukupno 25 stanja

Σ - konačan skup ulaznih znakova = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., x\}$

Γ - konačan skup znakova trake = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., x, _ \}$

q_0 - početno stanje = q_0

B - znak prazne ćelije = $_$

F - skup prihvatljivih stanja = $\{q_{24}\}$

δ - funkcija prijelaza je oblika: $\delta(q, (a,b,c)) \rightarrow (q_{\text{novo}}, (x, y, z), (P_1, P_2, P_3))$

gdje je:

q = stanje u kojem se trenutno nalazi

a = pročitani znak na prvoj traci

b = pročitani znak na drugoj traci

c = pročitani znak na trećoj traci

q_{novo} = novo stanje u koje upravljačka jedinka prelazi

x = znak koji se zapisuje na prvu traku

y = znak koji se zapisuje na drugu traku

z = znak koji se zapisuje na treću traku

$P_i \in \{r, l, s\}$ = određuje pomak za svaku glavu, r - desno, l - lijevo, s - ne miče se

Funkcije prijelaza

Faza 1 – pretvaranje IP adrese iz dekadskog u binarni oblik

Pretvaranje IP adrese iz dekadskog u binarni oblik bit će riješeno jednostavnim algoritmom. Ideja je da se dekadskom broju konstantno oduzima vrijednost 1 i pritom binarnom dodaje vrijednost 1. Taj postupak se ponavlja sve dok dekadski zapis ne dođe do nule. Tada će binarni zapis biti jednak početnoj dekadskoj vrijednosti. Binarni broj će u početku biti postavljen kao osam nula na koje će se zatim dodavati vrijednost 1.

U početku ulazni niz je zapisan na prvu traku i položaj glave je na prvoj znamenki slijeva. Kako bi pretvaranje moglo započeti najprije treba glavu pomaknuti do prve točke. Na kraju će se na mjestu točke zapisati znak x kako bi se ta točka razlikovala od ostalih točaka u zapisu. Funkcije prijelaza su:

$$\delta(q_0, (0, _, _)) \rightarrow (q_0, (0, _, _), (r,s,s))$$

$$\delta(q_0, (1, _, _)) \rightarrow (q_0, (1, _, _), (r,s,s))$$

$$\delta(q_0, (2, _, _)) \rightarrow (q_0, (2, _, _), (r,s,s))$$

$$\delta(q_0, (3, _, _)) \rightarrow (q_0, (3, _, _), (r,s,s))$$

$$\delta(q_0, (4, _, _)) \rightarrow (q_0, (4, _, _), (r,s,s))$$

$$\delta(q_0, (5, _, _)) \rightarrow (q_0, (5, _, _), (r,s,s))$$

$$\delta(q_0, (6, _, _)) \rightarrow (q_0, (6, _, _), (r,s,s))$$

$$\delta(q_0, (7, _, _)) \rightarrow (q_0, (7, _, _), (r,s,s))$$

$$\delta(q_0, (8, _, _)) \rightarrow (q_0, (8, _, _), (r,s,s))$$

$$\delta(q_0, (9, _, _)) \rightarrow (q_0, (9, _, _), (r,s,s))$$

$$\delta(q_0, (., _, _)) \rightarrow (q_1, (x, _, _), (s,s,s))$$

Sada je potrebno pripremiti drugu traku za dodavanje binarne jedinice. Svaki od četiri dijela neke IP adrese ima raspon od 0 do 255 što je zapravo raspon koji je moguće prikazati sa 8-bitnim binarnim brojem. Prema tome cilj je pomaknuti glavu za osam mjesta u desno i pritom tih osam mjesta ispuniti nulom kako bi zbrajanje uvijek mogli započeti od znamenke sa najmanjom težinom. Kako bi se to ostvarilo na treću traku bit će zapisan broj 8 koji će se postupno smanjivati za jedan nakon svakog pomaka u drugoj traci. Funkcije prijelaza su:

$$\begin{aligned}
\delta(q_1, (x, _, _)) &\rightarrow (q_1, (x, _, 8), (s,s,s)) \\
\delta(q_1, (x, _, 8)) &\rightarrow (q_1, (x, 0, 7), (s,r,s)) \\
\delta(q_1, (x, _, 7)) &\rightarrow (q_1, (x, 0, 6), (s,r,s)) \\
\delta(q_1, (x, _, 6)) &\rightarrow (q_1, (x, 0, 5), (s,r,s)) \\
\delta(q_1, (x, _, 5)) &\rightarrow (q_1, (x, 0, 4), (s,r,s)) \\
\delta(q_1, (x, _, 4)) &\rightarrow (q_1, (x, 0, 3), (s,r,s)) \\
\delta(q_1, (x, _, 3)) &\rightarrow (q_1, (x, 0, 2), (s,r,s)) \\
\delta(q_1, (x, _, 2)) &\rightarrow (q_1, (x, 0, 1), (s,r,s)) \\
\delta(q_1, (x, _, 1)) &\rightarrow (q_2, (x, 0, _), (l,r,s))
\end{aligned}$$

Sada kada je druga traka spremna, petlja oduzimanja jedinice dekadskom zapisu te zatim dodavanje jedinice binarnom zapisu može započeti. Najprije će od dekadskog zapisa oduzeti znamenku 1. Funkcije prijelaza su:

$$\begin{aligned}
\delta(q_2, (9, _, _)) &\rightarrow (q_3, (8, _, _), (r,s,s)) \\
\delta(q_2, (8, _, _)) &\rightarrow (q_3, (7, _, _), (r,s,s)) \\
\delta(q_2, (7, _, _)) &\rightarrow (q_3, (6, _, _), (r,s,s)) \\
\delta(q_2, (6, _, _)) &\rightarrow (q_3, (5, _, _), (r,s,s)) \\
\delta(q_2, (5, _, _)) &\rightarrow (q_3, (4, _, _), (r,s,s)) \\
\delta(q_2, (4, _, _)) &\rightarrow (q_3, (3, _, _), (r,s,s)) \\
\delta(q_2, (3, _, _)) &\rightarrow (q_3, (2, _, _), (r,s,s)) \\
\delta(q_2, (2, _, _)) &\rightarrow (q_3, (1, _, _), (r,s,s)) \\
\delta(q_2, (1, _, _)) &\rightarrow (q_3, (0, _, _), (r,s,s))
\end{aligned}$$

Ovim funkcijama ostvareno je oduzimanje za sve slučajeve osim kada je zadnja znamenka nula. U tom slučaju potrebno je pomaknuti glavu lijevo te oduzeti vrijednost 1 idućoj znamenki a trenutnu postaviti na 9.

$$\delta(q_2, (0, _, _)) \rightarrow (q_2, (0, _, _), (l,s,s))$$

$$\delta(q_3, (0, _, _)) \rightarrow (q_3, (9, _, _), (r,s,s))$$

Idući korak koji se ponavlja nakon svakog dekadskog oduzimanja je binarno zbrajanje sa znamenkom 1. U stanju q_3 oduzimanje je završeno i na red dolazi zbrajanje. Ako je zadnja znamenka 0 onda se na njenom mjestu zapiše 1 i zbrajanje je gotovo prelaskom u stanje q_4 . Ako je situacija suprotna i zadnja znamenka ima vrijednost 1 onda ćemo zapisati 0 i pomaknuti se lijevo kako bi idućoj znamenki dodali 1. To ostvarujemo tako da ostajemo u stanju q_3 sve dok ne dođemo do znamenke 0.

$$\delta(q_3, (x, _, _)) \rightarrow (q_3, (x, _, _), (s,l,s))$$

$$\delta(q_3, (x, 0, _)) \rightarrow (q_4, (x, 1, _), (s,r,s))$$

$$\delta(q_3, (x, 1, _)) \rightarrow (q_3, (x, 0, _), (s,l,s))$$

Zbrajanje je gotovo i sada je potrebno samo pomaknuti glavu druge trake nazad na kraj zapisa, tj. na prvu praznu ćeliju iza zadnjeg znaka. Nakon toga oduzimanje iznova može započeti prebacivanjem u stanje q_2 .

$$\delta(q_4, (x, 0, _)) \rightarrow (q_4, (x, 0, _), (s,r,s))$$

$$\delta(q_4, (x, 1, _)) \rightarrow (q_4, (x, 1, _), (s,r,s))$$

$$\delta(q_4, (x, _, _)) \rightarrow (q_2, (x, _, _), (l,s,s))$$

Prethodna dva bloka prijelaza će se izmjenjivati sve dok ne oduzmemo dekadski broj do nule. Kako bi se rad stroja nastavio moramo ga uhvatiti u tom trenutku. Dakle ako ostane zapis oblika $000x$ glava će se pomaknuti skroz lijevo do prve prazne ćelije pa se tako zna da je oduzimanje došlo do kraja. Stanje će biti prebačeno iz q_2 u q_0 i na drugu traku će biti zapisan x na kraju binarnog zapisa da bude moguće lakše prepoznati četiri bloka zapisa na kraju. Potreban je još jedan dodatni prijelaz kako bi u stanju q_0 glava mogla preskočiti preko znaka x te doći do

iduće točke. Ovaj proces se ponavlja sve dok ne prođe kroz sva četiri bloka zapisa IP adrese.

$$\delta(q_2, (_, _, _)) \rightarrow (q_0, (_, x, _), (r,r,s))$$

$$\delta(q_0, (x, _, _)) \rightarrow (q_0, (_, _, _), (r,s,s))$$

Kraj zadnjeg broja nije označen sa točkom već sa znakom /. Zato treba napisati prijelaz koji će prepoznati čitanje kose crte te na njeno mjesto zapisati x i pokrenuti oduzimanje prelaskom u stanje q1. Na kraju kada

$$\delta(q_0, (/ , _, _)) \rightarrow (q_1, (x, _, _), (s,s,s))$$

Kada više ne bude znakova . i / na putu znači da će glava doći do kraja niza i da je postupak pretvaranja završen. Po završetku postupka stroj će se prebaciti u novo stanje q5. Pretvaranje adrese iz dekadskog u binarni oblik je sada završeno.

$$\delta(q_0, (_, _, _)) \rightarrow (q_5, (_, _, _), (s,l,s))$$

Faza 2 – zapisivanje mrežne maske

Ideja oko zapisivanje mrežne maske na treću traku je da se najprije traka ispunji sa 32 nule odvojene u 4 bloka znakom x. To je potrebno kako bi format zapisa bio identičan binarnom obliku IP adrese. Zatim će preko upisanih nula, počevši od prve, biti upisano N jedinica. Upisivanje N jedinica riješeno je smanjivanjem broja N za vrijednost 1 tijekom svakog upisivanja. Kada se N oduzme do nule posao je gotov.

Postupak će započeti vraćanjem glave druge trake na početak zapisa. Kraj prethodne faze označen je prelaskom u stanje q5 pa ćemo tu i nastaviti. Nakon pomicanja upravljačka jedinka će preći u novo stanje q6.

$$\delta(q_5, (_, x, _)) \rightarrow (q_5, (_, x, _), (s,l,s))$$

$$\delta(q_5, (_, 0, _)) \rightarrow (q_5, (_, 0, _), (s,l,s))$$

$$\delta(q_5, (_, 1, _)) \rightarrow (q_5, (_, 1, _), (s,l,s))$$

$$\delta(q_5, (_, _, _)) \rightarrow (q_6, (_, _, _), (s,r,s))$$

U stanju q6 cilj je popuniti treću traku nulama. Pritom će glava druge trake prolaziti kroz svoj 32-bitni zapis i tako će stroj znati kada treba zapisati 0, kada x, a naposljetku i kada treba stati.

$$\delta(q6, (_, 0, _)) \rightarrow (q6, (_, 0, 0), (s,r,r))$$

$$\delta(q6, (_, 1, _)) \rightarrow (q6, (_, 1, 0), (s,r,r))$$

$$\delta(q6, (_, x, _)) \rightarrow (q6, (_, x, x), (s,r,r))$$

$$\delta(q6, (_, _, _)) \rightarrow (q7, (_, _, _), (s,l,l))$$

U stanju q7 glave druge i treće trake pomaknuti će se na početke svojih zapisa. Prikazani su prijelaze za sve kombinacije znakova druge i treće trake. Kada glave budu u položajima ispred prvih znakova zapisa treću glava će se pomaknuti na prvi znak. Pritom će i prva glava biti pomaknuta na zadnji znak broja N kako bi oduzimanje i zapisivanje jedinica moglo započeti u novom stanju q8.

$$\delta(q7, (_, x, x)) \rightarrow (q7, (_, x, x), (s,l,l))$$

$$\delta(q7, (_, 0, 0)) \rightarrow (q7, (_, 0, 0), (s,l,l))$$

$$\delta(q7, (_, 1, 0)) \rightarrow (q7, (_, 1, 0), (s,l,l))$$

$$\delta(q7, (_, _, _)) \rightarrow (q8, (_, _, _), (l,s,r))$$

Sada započinje upisivanje N jedinica u treću traku. Oduzima se znamenka 1 od N i pritom zapisuje jedna jedinica. Evo prijelaza za slučajeve kad zadnja znamenka nije 0.

$$\delta(q8, (1, _, 0)) \rightarrow (q8, (0, _, 1), (s,s,r))$$

$$\delta(q8, (2, _, 0)) \rightarrow (q8, (1, _, 1), (s,s,r))$$

$$\delta(q8, (3, _, 0)) \rightarrow (q8, (2, _, 1), (s,s,r))$$

$$\delta(q8, (4, _, 0)) \rightarrow (q8, (3, _, 1), (s,s,r))$$

$$\delta(q8, (5, _, 0)) \rightarrow (q8, (4, _, 1), (s,s,r))$$

$$\delta(q8, (6, _, 0)) \rightarrow (q8, (5, _, 1), (s,s,r))$$

$$\delta(q8, (7, _, 0)) \rightarrow (q8, (6, _, 1), (s,s,r))$$

$$\delta(q8, (8, _, 0)) \rightarrow (q8, (7, _, 1), (s,s,r))$$

$$\delta(q8, (9, _, 0)) \rightarrow (q8, (8, _, 1), (s,s,r))$$

Ako je zadnja znamenka 0 onda glava treba zapisati 9, otići na iduću znamenku lijevo, oduzeti joj 1 i vratiti se nazad. Budući da je broj N manji ili jednak broju 32 nije potrebno ostvarivati univerzalno oduzimanje za više znamenaka no što je potrebno. Na kraju upravljačka jedinka prelazi u stanje q10 koje označava kraj zapisivanja jedinica u mrežnu masku.

$$\begin{aligned}\delta(q8, (0, _, 0)) &\rightarrow (q9, (9, _, 0), (l,s,s)) \\ \delta(q9, (3, _, 0)) &\rightarrow (q8, (2, _, 1), (r,s,r)) \\ \delta(q9, (2, _, 0)) &\rightarrow (q8, (1, _, 1), (r,s,r)) \\ \delta(q9, (1, _, 0)) &\rightarrow (q8, (0, _, 1), (r,s,r)) \\ \delta(q9, (0, _, 0)) &\rightarrow (q9, (0, _, 0), (l,s,s)) \\ \delta(q9, (_, _, 0)) &\rightarrow (q10, (_, _, 0), (r,s,s))\end{aligned}$$

Prilikom zapisivanja jedinica na treću traku potrebno je paziti da glava preskoči sve znakove x u zapisu. Sljedeće funkcije prijelaza to omogućuju:

$$\begin{aligned}\delta(q8, (0, _, x)) &\rightarrow (q8, (0, _, x), (s,s,r)) \\ \delta(q8, (1, _, x)) &\rightarrow (q8, (1, _, x), (s,s,r)) \\ \delta(q8, (2, _, x)) &\rightarrow (q8, (2, _, x), (s,s,r)) \\ \delta(q8, (3, _, x)) &\rightarrow (q8, (3, _, x), (s,s,r)) \\ \delta(q8, (4, _, x)) &\rightarrow (q8, (4, _, x), (s,s,r)) \\ \delta(q8, (5, _, x)) &\rightarrow (q8, (5, _, x), (s,s,r)) \\ \delta(q8, (6, _, x)) &\rightarrow (q8, (6, _, x), (s,s,r)) \\ \delta(q8, (7, _, x)) &\rightarrow (q8, (7, _, x), (s,s,r)) \\ \delta(q8, (8, _, x)) &\rightarrow (q8, (8, _, x), (s,s,r)) \\ \delta(q8, (9, _, x)) &\rightarrow (q8, (9, _, x), (s,s,r))\end{aligned}$$

Postoji jedna iznimka za slučaj kada je $N = 32$. Tada glava treće trake na kraju zapisivanja neće ostati na znaku 0 već će preći na prvu praznu ćeliju iza zapisa. Zato je prioritet osigurati da se u tom iznimnom slučaju glava pomakne jedno mjesto u lijevo kako bi ostala na nuli kao što je slučaj za sve ostale mogućnosti.

$$\delta(q8, (0, _, _)) \rightarrow (q10, (0, _, _), (r,s,l))$$

Faza 3 – logička operacija I nad zapisom IP adrese i mrežnom maskom

Cilj u ovoj fazi je redom prolaziti znakove zapisa u drugoj i trećoj traci i obavljati logičku I (AND) operaciju nad bitovima. U prethodnoj fazi stali smo u stanju q10. Na početku se glava prve trake pomiče skroz desno na prazne ćelije kako bi se mogli početi zapisivati rezultati logičke operacije.

$$\delta(q_{10}, (0, _, 0)) \rightarrow (q_{10}, (_, _, 0), (r,s,s))$$

$$\delta(q_{10}, (9, _, 0)) \rightarrow (q_{10}, (_, _, 0), (r,s,s))$$

$$\delta(q_{10}, (_, _, 0)) \rightarrow (q_{11}, (_, _, 0), (r,s,s))$$

$$\delta(q_{10}, (_, _, x)) \rightarrow (q_{11}, (_, _, x), (r,s,s))$$

Da bi izvršavanje logičke operacije moglo započeti, potrebno je najprije vratiti glavu treće trake na početak niza te ući u novo q12 stanje koje će aktivirati početak računanja.

$$\delta(q_{11}, (_, _, 0)) \rightarrow (q_{11}, (_, _, 0), (s,s,l))$$

$$\delta(q_{11}, (_, _, 1)) \rightarrow (q_{11}, (_, _, 1), (s,s,l))$$

$$\delta(q_{11}, (_, _, x)) \rightarrow (q_{11}, (_, _, x), (s,s,l))$$

$$\delta(q_{11}, (_, _, _)) \rightarrow (q_{12}, (_, _, _), (s,r,r))$$

Sada je sve spremno i računanje logičke operacije I može započeti. Dane su funkcije prijelaza za sve moguće ulazne kombinacije, a njihova rješenja logičke operacije zapisuju se u prvu traku.

$$\delta(q_{12}, (_, 0, 0)) \rightarrow (q_{12}, (0, 0, 0), (r,r,r))$$

$$\delta(q_{12}, (_, 0, 1)) \rightarrow (q_{12}, (0, 0, 1), (r,r,r))$$

$$\delta(q_{12}, (_, 1, 0)) \rightarrow (q_{12}, (0, 1, 0), (r,r,r))$$

$$\delta(q_{12}, (_, 1, 1)) \rightarrow (q_{12}, (1, 1, 1), (r,r,r))$$

$$\delta(q_{12}, (_, x, x)) \rightarrow (q_{12}, (x, x, x), (r,r,r))$$

Na kraju računanja upravljačka jedinka prelazi u novo stanje kojim započinje posljednja četvrta faza.

$$\delta(q_{12}, (_, _, _)) \rightarrow (q_{13}, (_, _, _), (l,s,r))$$

Faza 4 – pretvaranje binarne vrijednosti IP adrese pod mreže u . notaciju

U prethodnoj fazi logičkom operacijom I dobivena je binarna vrijednost IP adrese pod mreže. Adresa je zapisana u prvoj traci. Ideja oko pretvorbe iz binarnog u dekadski oblik je analogna ideji obrnutog pretvaranja u prvoj fazi. Binarnom broju će se konstantno oduzimati vrijednost 1 i pritom dekadskom dodavati oduzeta vrijednost. Dekadski broj će u startu imati vrijednost 0, a pretvaranje je završeno kada vrijednost binarnog zapisa bude 0.

Prilikom pretvaranja IP adresa u dekadskom obliku će se zapisivati na treću traku pa će se glava prije morati pomaknuti u desno za 16 mjesta kako bi bilo dovoljno prostora za spremanje svih znamenaka. Potrebno je 16 mjesta jer će se pretvaranje izvoditi od zadnjeg dijela IP adrese prema prvom. Dakle zapisivanje će se izvoditi s desna na lijevo. Pomicanje za 16 mjesta može se ostvariti zapisivanjem broja 16 koji će poslužiti kao brojač prilikom pomicanja glave. U ovom slučaju može se puno jednostavnije izvesti ako se iskoristiti oblik zapisa IP adrese gdje se 8 binarnih znakova nalazi između dva znaka x. Dakle potrebno je samo napisati funkcije prijelaza koje će dva puta proći kroz jedan takav blok zapisa i pritom pomicati glavu treće trake u desno. Sljedeće funkcije prijelaza upravo to i rade. Pomiču glavu prve trake do predzadnjeg znaka x pa se zatim vraćaju nazad do zadnjeg što čini ukupno 16 pomaka po binarnom zapisu. Zadnji prijelaz na kraju stavlja točku koja označava kraj pojedinog bloka u . notaciji.

$$\delta(q_{13}, (x, _, _)) \rightarrow (q_{14}, (x, _, _), (l, s, s))$$

$$\delta(q_{14}, (0, _, _)) \rightarrow (q_{14}, (0, _, _), (l, s, r))$$

$$\delta(q_{14}, (1, _, _)) \rightarrow (q_{14}, (1, _, _), (l, s, r))$$

$$\delta(q_{14}, (x, _, _)) \rightarrow (q_{15}, (x, _, _), (r, s, s))$$

$$\delta(q_{15}, (0, _, _)) \rightarrow (q_{15}, (0, _, _), (r, s, r))$$

$$\delta(q_{15}, (1, _, _)) \rightarrow (q_{15}, (1, _, _), (r, s, r))$$

$$\delta(q_{15}, (x, _, _)) \rightarrow (q_{16}, (x, _, .), (s, s, s))$$

Pretvaranje sada može započeti. Najprije će se provoditi oduzimanje vrijednosti 1 binarnom zapisu.

$$\delta(q_{16}, (x, _, .)) \rightarrow (q_{17}, (x, _, .), (l, s, s))$$

$$\delta(q_{17}, (0, _, .)) \rightarrow (q_{17}, (1, _, .), (l, s, s))$$

$$\delta(q_{17}, (1, _, .)) \rightarrow (q_{18}, (0, _, .), (r, s, s))$$

Nakon oduzimanja potrebno je glavu prve trake vratiti na početak niza tj. do prvog znaka x u desno. To će biti učinjeno u stanju q_{18} .

$$\delta(q_{18}, (1, _, .)) \rightarrow (q_{18}, (1, _, .), (r, s, s))$$

$$\delta(q_{18}, (0, _, .)) \rightarrow (q_{18}, (0, _, .), (r, s, s))$$

$$\delta(q_{18}, (x, _, .)) \rightarrow (q_{19}, (x, _, .), (s, s, l))$$

Nakon vraćanja glave na početak niza može se dodati vrijednost 1 dekadskom zapisu. To je ostvareno u stanju q_{19} . Treba primijetiti da se funkcije prijelaza za slučaj kad je zadnja znamenka 9 opet definiraju zasebno kako bi se prenosio ostatak na idući broj.

$$\delta(q_{19}, (x, _, .)) \rightarrow (q_{21}, (x, _, .), (s, s, l))$$

$$\delta(q_{19}, (x, _, _)) \rightarrow (q_{21}, (x, _, 1), (s, s, r))$$

$$\delta(q_{19}, (x, _, 0)) \rightarrow (q_{21}, (x, _, 1), (s, s, r))$$

$$\delta(q_{19}, (x, _, 1)) \rightarrow (q_{21}, (x, _, 2), (s, s, r))$$

$$\delta(q_{19}, (x, _, 2)) \rightarrow (q_{21}, (x, _, 3), (s, s, r))$$

$$\delta(q_{19}, (x, _, 3)) \rightarrow (q_{21}, (x, _, 4), (s, s, r))$$

$$\delta(q_{19}, (x, _, 4)) \rightarrow (q_{21}, (x, _, 5), (s, s, r))$$

$$\delta(q_{19}, (x, _, 5)) \rightarrow (q_{21}, (x, _, 6), (s, s, r))$$

$$\delta(q_{19}, (x, _, 6)) \rightarrow (q_{21}, (x, _, 7), (s, s, r))$$

$$\delta(q_{19}, (x, _, 7)) \rightarrow (q_{21}, (x, _, 8), (s, s, r))$$

$$\delta(q_{19}, (x, _, 8)) \rightarrow (q_{21}, (x, _, 9), (s, s, r))$$

$$\delta(q_{21}, (x, _, .)) \rightarrow (q_{16}, (x, _, .), (s, s, s))$$

U slučaju da je zadnja znamenka 9 potrebno je na njeno mjesto zapisati 0 i idućoj znamenki dodati 1. Prijelazi su ostvareni tako da se glava najprije pomakne lijevo do broja koji nije 9 i doda mu 1, a zatim se vraća istim putem prepisujući sve znakove 9 sa nulom.

$$\delta(q_{19}, (x, _, 9)) \rightarrow (q_{19}, (x, _, 9), (s,s,l))$$

$$\delta(q_{21}, (x, _, 9)) \rightarrow (q_{21}, (x, _, 0), (s,s,r))$$

Kada ovaj način pretvaranja dođe do kraja stroj se mora prebaciti na idući blok i ponovno pozvati algoritam oduzimanja i zbrajanja. Najprije moramo dodati novu točku u trećoj traci lijevo od zadnjeg pretvorenog bloka. Dakle svi brojevi ispred trenutne točke se preskaču i na početak se stavlja nova točka.

$$\delta(q_{17}, (x, _, .)) \rightarrow (q_{20}, (x, _, .), (s,s,l))$$

$$\delta(q_{20}, (x, _, 0)) \rightarrow (q_{20}, (x, _, 0), (s,s,l))$$

$$\delta(q_{20}, (x, _, 1)) \rightarrow (q_{20}, (x, _, 1), (s,s,l))$$

$$\delta(q_{20}, (x, _, 2)) \rightarrow (q_{20}, (x, _, 2), (s,s,l))$$

$$\delta(q_{20}, (x, _, 3)) \rightarrow (q_{20}, (x, _, 3), (s,s,l))$$

$$\delta(q_{20}, (x, _, 4)) \rightarrow (q_{20}, (x, _, 4), (s,s,l))$$

$$\delta(q_{20}, (x, _, 5)) \rightarrow (q_{20}, (x, _, 5), (s,s,l))$$

$$\delta(q_{20}, (x, _, 6)) \rightarrow (q_{20}, (x, _, 6), (s,s,l))$$

$$\delta(q_{20}, (x, _, 7)) \rightarrow (q_{20}, (x, _, 7), (s,s,l))$$

$$\delta(q_{20}, (x, _, 8)) \rightarrow (q_{20}, (x, _, 8), (s,s,l))$$

$$\delta(q_{20}, (x, _, 9)) \rightarrow (q_{20}, (x, _, 9), (s,s,l))$$

$$\delta(q_{20}, (x, _, _)) \rightarrow (q_{22}, (x, _, .), (s,s,r))$$

U prethodnom dijelu zadnja napisana funkcija prijelaza koja zapisuje novu točku pomiče treću glavu nazad na desni blok brojeva te prelazi u potpuno novo stanje. Budući da to izgleda poprilično nerazumljivo u nastavku je detaljno objašnjeno zašto je to nužno.

Postoji jedna iznimka koju je potrebno uključiti u funkcije prijelaza. Ako su sve znamenke u bloku binarnog zapisa na kojem se izvodi pretvaranje u samom početku jednake nulama, onda će prema trenutnim funkcijama prijelaza glava preći na idući blok i tako u dekadskom zapisu ostaviti _____. Umjesto _____.0.

Dakle treba napisati funkcije koje će obuhvatiti i riješiti problem izazvan ovim iznimnim slučajem. To će biti napravljeno tako da će se u trenutku kada svi znakovi u binarnom zapisu budu imali vrijednost 0, dakle na samom kraju pretvaranja tog binarnog bloka, nakon što je lijevo od zapisa upisana nova točka, provjeriti da li ispred točke već postoji neki broj ili je u pitanju prethodna točka. Ako je točka ispred točke onda će se zapisati znamenka 0 između njih i postupak se može nastaviti. Korišteno je zapisivanje u drugu traku na principu brojača kako bi se drugi i treći prijelaz izvršili točno jedan za drugim. Time se izbjeglo stvaranje dodatnih stanja. Na kraju ako je nula ubačena može se prebaciti u stanje q16 i započeti pretvaranje idućeg bloka.

$$\delta(q_{22}, (x, _, .)) \rightarrow (q_{22}, (x, 1, .), (s, s, l))$$

$$\delta(q_{22}, (x, 1, .)) \rightarrow (q_{22}, (x, 0, 0), (s, s, l))$$

$$\delta(q_{22}, (x, 0, _)) \rightarrow (q_{16}, (x, _, .), (s, s, s))$$

Također je nužno napisati i prijelaze za slučaj ako je ispred točke upisan broj kako stroj ne bi stao u svim slučajevima kada binarni blok koji pretvaramo u dekadski ima vrijednost veću od nule.

$$\delta(q_{22}, (x, _, 0)) \rightarrow (q_{16}, (x, _, 0), (s, s, l))$$

$$\delta(q_{22}, (x, _, 1)) \rightarrow (q_{16}, (x, _, 1), (s, s, l))$$

$$\delta(q_{22}, (x, _, 2)) \rightarrow (q_{16}, (x, _, 2), (s, s, l))$$

$$\delta(q_{22}, (x, _, 3)) \rightarrow (q_{16}, (x, _, 3), (s, s, l))$$

$$\delta(q_{22}, (x, _, 4)) \rightarrow (q_{16}, (x, _, 4), (s, s, l))$$

$$\delta(q_{22}, (x, _, 5)) \rightarrow (q_{16}, (x, _, 5), (s, s, l))$$

$$\delta(q_{22}, (x, _, 6)) \rightarrow (q_{16}, (x, _, 6), (s, s, l))$$

$$\delta(q_{22}, (x, _, 7)) \rightarrow (q_{16}, (x, _, 7), (s, s, l))$$

$$\delta(q_{22}, (x, _, 8)) \rightarrow (q_{16}, (x, _, 8), (s, s, l))$$

$$\delta(q_{22}, (x, _, 9)) \rightarrow (q_{16}, (x, _, 9), (s, s, l))$$

Na kraju posla dogodit će se da će u stanju q17 glava prve trake doći na prvu praznu ćeliju lijevo od zapisa. Taj događaj bit će uhvaćen idućom funkcijom prijelaza i pritom će stroj preći u novo stanje koje će označiti kraj pretvorbe.

$$\delta(q_{17}, (_, _, .)) \rightarrow (q_{23}, (_, _, .), (s, s, s))$$

Na samom kraju u trećoj traci dobit će se IP adresa pod mreže u xxx.xxx.xxx.xxx. formatu. Iz estetskih razloga može se ukloniti posljednju točku budući da je sada nepotrebna u zapisu. Da bi to obavila glava se kreće desno kroz sve znakove dok ne dođe do prve prazne ćelije. Tada se glava vraća jedno mjesto u lijevo i briše zadnju točku. Upravljačka jedinka na samom kraju završava u stanju q24 pa je to stanje označeno kao prihvatljivo.

$$\begin{aligned} \delta(q_{23}, (_, _, .)) &\rightarrow (q_{23}, (_, _, .), (s,s,r)) \\ \delta(q_{23}, (_, _, 0)) &\rightarrow (q_{23}, (_, _, 0), (s,s,r)) \\ \delta(q_{23}, (_, _, 1)) &\rightarrow (q_{23}, (_, _, 1), (s,s,r)) \\ \delta(q_{23}, (_, _, 2)) &\rightarrow (q_{23}, (_, _, 2), (s,s,r)) \\ \delta(q_{23}, (_, _, 3)) &\rightarrow (q_{23}, (_, _, 3), (s,s,r)) \\ \delta(q_{23}, (_, _, 4)) &\rightarrow (q_{23}, (_, _, 4), (s,s,r)) \\ \delta(q_{23}, (_, _, 5)) &\rightarrow (q_{23}, (_, _, 5), (s,s,r)) \\ \delta(q_{23}, (_, _, 6)) &\rightarrow (q_{23}, (_, _, 6), (s,s,r)) \\ \delta(q_{23}, (_, _, 7)) &\rightarrow (q_{23}, (_, _, 7), (s,s,r)) \\ \delta(q_{23}, (_, _, 8)) &\rightarrow (q_{23}, (_, _, 8), (s,s,r)) \\ \delta(q_{23}, (_, _, 9)) &\rightarrow (q_{23}, (_, _, 9), (s,s,r)) \\ \delta(q_{23}, (_, _, _)) &\rightarrow (q_{24}, (_, _, _), (s,s,l)) \\ \delta(q_{24}, (_, _, .)) &\rightarrow (q_{24}, (_, _, _), (s,s,s)) \end{aligned}$$

Zaključak

U ovom seminarskom radu opisan je postupak određivanja IP adrese pod mreže na osnovu zadane IP adrese i njene maske. Najveći dio ostvarivanja postupka sveo se na pretvaranje iz dekadskih u binarni oblik u početku te vraćanje iz binarnog u dekadski na samom kraju. Dva algoritma za izvođenje ova dva koraka su bila osmišljena i razmatrana. Prvi algoritam, ujedno i odabrani, temelji se na uzastopnom oduzimanju vrijednosti 1 broju kojeg pretvaramo te istovremeno dodavanje vrijednosti 1 broju u brojevnom sustavu koji se mora dobiti. Drugi razmatrani algoritam je bila ideja ostvarivanja konstantnog dijeljenja dekadskog broja sa brojem 2 te zapisivanja ostatka što je i općeprihvaćeni način pretvaranja dekadskog broja u binarni. No onda bi se za vraćanje iz binarnog u dekadski oblik moralo ostvariti zbrajanje potencija broja 2. Dakako da je drugi algoritam puno složeniji i samim time bi zahtijevao poprilično veći broj funkcija prijelaza i stanja, što bi ukupnu konstrukciju stroja samo zakompliciralo. Jedina prednost drugog algoritma je manji broj koraka prilikom rada stroja. No za razumljivost konstruiranog stroja puno je važnije da ima što manje prijelaza i stanja nego manje koraka tijekom izvođenja. Iz tih razloga odabran je prvi algoritam i pokazao se kao dobar izbor.

Za konstrukciju Turingovog stroja korišten je prošireni model sa višestrukim trakama. TS identičan ovome može se izgraditi i sa samo jednom trakom jer su svi prošireni modeli TS jednako moćni kao osnovni model. No ostvarenje putem stroja sa jednom trakom bilo bi mnogo složenije i kompliciranije. Uzevši to u obzir, tri trake su bile najoptimalniji izbor.

Uz seminarski rad priložen je i simulator Turingovog stroja sa definicijom stroja ostvarenog u ovom radu. Definicija je spremljena u tekstualnoj datoteci i u njoj se nalaze sve prethodno opisane funkcije prijelaza na jednom mjestu. Uloga simulatora je bila dodana pomoć pri testiranju rada stroja, a može poslužiti i kao pomoć pri razumijevanju rada stroja.