

Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva

Vanja Smailović

**Seminarski rad iz predmeta  
Uvod u teoriju računarstva**

Zadatak broj 3040.

Zagreb, lipanj 2008.

# **Seminarski rad iz predmeta Uvod u teoriju računarstva**

**Student:** Vanja Smailović

**Matični broj studenta:** 0036 434023

## **Zadatak broj 3040:**

Napisati program za simulaciju rada Turingovog stroja koji provjerava da li su dva dekadski broja djeljiva. Format zapisa na traci je: broj1/broj2=

# Uvod

Turingov se sastoji od upravljačke jedinice koja uz pomoć glave za čitanje i pisanje čita odnosno zapisuje znakove sa ulazne trake, koja ima krajnje lijevu ćeliju dok je sa desne strane neograničena, te mijenja stanja. Na početku rada n krajnje lijevih ćelija sadrže niz  $w$  gdje je  $|w| = n$ , a  $n \geq 0$ , znakovi tog niza i znakovi koje TS zapisuje na traku čine skup znakova trake. Ostatak trake je popunjen praznim ćelijama koje se označavaju sa  $B$ . Tijekom rada na temelju pročitano znaka i trenutnog stanja upravljačke jedinice TS odlučuje: u koje novo stanje jedinka prelazi, koji znak se zapisuje na traku umjesto pročitano znaka i u koju se stranu pomiče glava za čitanje i pisanje.

TS se zadaje kao uređena sedmorka:

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

gdje je

$Q$  – konačan skup stanja

$\Gamma$  – konačan skup znakova trake

$B$  - znak kojim se označava prazna ćelija

$\Sigma$  – konačan skup ulaznih znakova

$\delta$  – funkcija prijelaza  $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$q_0$  – početno stanje

$F$  – skup prihvatljivih stanja

# Ostvarenje

U slučaju zadatka 3040., Turingov stroj izgleda ovako:

TS = ( {q<sub>0</sub>, q<sub>1</sub>, q<sub>2</sub>, q<sub>3</sub>, q<sub>4</sub>, q<sub>5</sub>, q<sub>6</sub>}, {P, 0, /, =}, {P, 0, /, =, X}, δ, q<sub>0</sub>, X, {q<sub>6</sub>} )

δ	P	0	/	=	X
q <sub>0</sub>	q <sub>1</sub> , P, R	-	-	-	-
q <sub>1</sub>	q <sub>5</sub> , P, R	q <sub>1</sub> , 0, R	q <sub>2</sub> , /, R	-	q <sub>1</sub> , X, R
q <sub>2</sub>	-	q <sub>3</sub> , X, L	-	q <sub>4</sub> , =, L	q <sub>2</sub> , X, R
q <sub>3</sub>	q <sub>5</sub> , P, R	q <sub>1</sub> , X, L	q <sub>3</sub> , /, L	-	q <sub>3</sub> , X, L
q <sub>4</sub>	-	-	q <sub>1</sub> , /, L	-	q <sub>4</sub> , 0, L
q <sub>5</sub>	-	-	q <sub>5</sub> , /, R	q <sub>6</sub> , =, L	q <sub>5</sub> , X, R

Osnovna ideja kako podijeliti dva broja ostvarena je preko modificiranog algoritma za uzastopno oduzimanje.

Početno stanje stroja je q<sub>0</sub>, a jedino prihvatljivo jest q<sub>6</sub>. Zapis na traci stroja je u obliku niza nula. Dakle, npr. P000/00= jest u biti izraz 3/2=.

## Uloge stanja i prijelaza

$$\delta(q_0, P) = (q_1, P, R)$$

Za početak, glava čita znak P, te se pomiče udesno da bi se došla do prvog broja. Stanje se mijenja u q<sub>1</sub>.

$$\delta(q_1, 0) = (q_1, 0, R)$$

$$\delta(q_1, X) = (q_1, X, R)$$

$$\delta(q_1, /) = (q_2, /, R)$$

Stanjem q<sub>1</sub> prelazimo preko cijelog prvo broja, da bismo došli do broja2, s kojim počinjemo algoritam uzastopnog oduzimanja. Kako smo pročitali znak /, prešli smo u stanje q<sub>2</sub> koje nam je potrebno za broj2.

$$\delta(q_1, P) = (q_5, P, R)$$

Ovaj prijelaz je trenutno van konteksta, no služi kako bismo prepoznali kraj algoritma za uzastopno oduzimanje. Ostvariti će se u slučaju da su svi znakovi 0 u broju1 pretvoreni u znakove X, te ih je glava čitajući došla do znaka P, oznake za početak trake. Stroj će iz stanja q<sub>1</sub> preći u q<sub>5</sub>, a prijelazi vezani za to stanje će biti opisani kasnije.

$$\begin{aligned}\delta(q_2, 0) &= (q_3, X, L) \\ \delta(q_2, X) &= (q_2, X, R) \\ \delta(q_2, =) &= (q_4, =, L)\end{aligned}$$

Ova skupina prijelaza je vezana za stanje  $q_2$ , dakle glava je uvijek negdje na broju2.

Prvi prijelaz će pretvoriti znak 0 u X, preći u novo stanje  $q_3$ , te pomaknuti glavu lijevo. Stanje  $q_3$  će biti opisano niže, a odnos stanja  $q_2$  i  $q_3$  je u principu algoritam uzastopnog oduzimanja, te se može opisati s: „ako se znak 0 pretvori u X u broju2 (stanje  $q_2$ ), tada se to isto mora učiniti i na broju1 (stanje  $q_3$ )”.

Drugi prijelaz preskače znakove X u broju2, iz razloga da se dođe do prve slijedeće nule.

Treći prijelaz je tu iz razloga što će, u slučaju da je  $\text{broj2} > \text{broj1}$ , kad-tad doći do situacije da je kompletan broj2 pretvoren u znakove X. U tom slučaju valja „resetirati” broj2, tako što ćemo ga vratiti u početno stanje. Stanje  $q_4$  će biti opisano niže.

$$\begin{aligned}\delta(q_3, 0) &= (q_1, X, L) \\ \delta(q_3, X) &= (q_3, X, L) \\ \delta(q_3, /) &= (q_3, /, L) \\ \delta(q_3, P) &= (q_5, P, R)\end{aligned}$$

Ova skupina prijelaza je vezana za stanje  $q_3$ . U stanje  $q_3$  se može preći samo čitanjem znaka 0 u stanju  $q_2$ . Stanje  $q_3$  je vezano za kretanje glave ulijevo, preko broja1.

Prvi prijelaz pretvara krajnje desni znak 0 broja1 u znak X, radi toga što je već isto učinjeno na broju2. Drugi korak algoritma za uzastopno oduzimanje je gotov, te se kreće iznova. Radi toga se prelazi u stanje  $q_1$ .

Drugi prijelaz jednostavno ignorira znakove X broja1, te se glava kreće lijevo sve do znaka 0.

Treći prijelaz je za slučaj da se glava našla na znaku / nakon prelaska iz stanje  $q_2$  u  $q_3$ .

Četvrti prijelaz je tu da bismo mogli prepoznati da je algoritam pri kraju, odnosno da je cijeli broj1 pretvoren u niz znakova X, tj. da je broj2 najveći mogući broj puta (cjelobrojno) oduzet od broja1. Radi ovoga prelazimo u stanje  $q_5$ , koje će biti opisano niže.

$$\begin{aligned}\delta(q_4, X) &= (q_4, 0, L) \\ \delta(q_4, /) &= (q_1, /, L)\end{aligned}$$

Ova skupina prijelaza je vezana za stanje  $q_4$ . Stanje  $q_4$  služi da bismo „resetirali” broj2, tj. vratili ga na početno stanje, jednom kada se kompletan broj2 nađe zapisan kao niz znakova X. Prvi prijelaz „vraća” znakove X u znakove 0, a drugi prijelaz služi da bi se znalo kada prestati s resetiranjem.

$$\begin{aligned}\delta(q_5, X) &= (q_5, X, R) \\ \delta(q_5, /) &= (q_5, /, R) \\ \delta(q_5, =) &= (q_6, =, L)\end{aligned}$$

U ovo stanje se prelazi u slučaju da je cijeli broj1 pretvoren u znakove X. Ovo stanje služi da bismo završili utvrđivanje djeljivosti brojeva. U slučaju da na traci više nema nula, glava će (kretajući se udesno) doći do znaka =. Preći će se u stanje  $q_6$ , te će se niz prihvatiti.

U slučaju da postoji ijedna nula na traci, stroj će stati, te se niz neće prihvatiti, jer u stanju  $q_5$  nemamo definiran prijelaz za znak 0 sa trake.

## Primjer

Algoritam preko kojeg je riješen zadatak ću objasniti kroz primjer P00000/00=, odnosno 5/2.

Glava čita krajnje lijevi znak P i prelazi u stanje q1.

**P00000/00=**

Stanje q1 služi za prelazak udesno preko broja1 i dolazak do znaka /. Kad se pročita znak /, prelazi se u stanje q2 i glava se pomiče desno, na broj2.

**P00000/00=**

U stanju q2 se traži prva 0, koja će se pretvoriti u X.

**P00000/X0=**

Princip po kojem je zamišljen algoritam jest da broj2 uzastopno oduzimamo od broja1 tako da za svaki znak 0 broja2, pretvorimo i jedan iz broja1 u znak X. Dakle, u stanju q2, kada se pročita 0 od broja2, pretvori se u X, te prelazi u stanje q3. Stanje q3 služi da bi se pomicalo ulijevo sve do prve 0, dakle krajnje desne, od broja1. Tu se, za onaj 0 iz broja2, jedna 0 pretvara u X.

**P0000X/X0=**

Nakon što smo to obavili, prelazimo opet u stanje q1, kojim prelazimo preko cijelog broja1 udesno, sve do znaka /. Stanjem q2 dolazimo do sljedeće 0 u broju2. Pretvaramo znak u X i pomičemo glavu ulijevo.

**P0000X/X0=**

**P0000X/XX=**

Stanjem q3 dolazimo do sljedeće 0 u broju1. Pretvaramo znak u X i pomičemo glavu ulijevo, prelazeći u stanje q1.

**P0000X/XX=**

**P000XX/XX=**

Kako smo prešli u stanje q1, opet ćemo pomicati glavu udesno sve dok ne dođemo do sljedeće 0 broja2. Kako glava čita samo znakove X u ovom slučaju, doći će do znaka =. Kada pročitaмо znak =, prelazimo u stanje q4. Stanje q4 je „resetirajuće” stanje, odnosno služi da bismo vratili broj2 u početno stanje. To znači da moramo XX prebaciti u znakove 00. Glava se pomiče ulijevo i svaki znak X pretvara u 0, sve do znaka /. Kada se u stanju q4 pročita znak /, vraćamo se u stanje q1 i algoritam počinje iznova, s tim da je broj 0 u broju1 smanjen na 3, uz dva znaka X desno.

**P000XX/XX=**

**P000XX/00=**

Pred kraj algoritma, stanje na traci je ovakvo:  
Automat je u stanju q2, te je 2. puta „resetirao” broj2.

**P0XXXX/00=**

Čita se znak 0 od broja2, pretvara u X.

**P0XXXX/X0=**

Glava se pomiče ulijevo, jer za upravo upisani X na broju2, moramo upisati isti i na broj1.  
Čitamo posljednju, krajnje lijevu nulu broja1, te ju pretvaramo u znak X, prelazeći u stanje q1 i pomičući glavu ulijevo, ovaj puta na znak P.

**PXXXXX/X0=**

Kad se u stanju q1 pročita znak P, prelazi se u novo stanje q5, obzirom da je ovo kraj algoritma, jer smo promijenili sve znakove 0 u X, kod broja1.

**PXXXXX/X0=**

U stanju q5 prelazimo preko svih X iz broja1, prelazimo preko / i dolazimo do broja2. Tu je ključ algoritma. Obzirom da 3 nije djeljiv sa 2, broj2 se neće „stići resetirati”, odnosno, neće imati samo znakove X, već i neki broj znakova 0.  
Zato ćemo, u stanju q5, pročitati taj znak 0 iz broja2, te će tu stroj stati, jer nema definiran prijelaz.

**PXXXXX/X0=** → TS staje, nema definiran prijelaz.

U slučaju da su brojevi bili djeljivi, kao u slučaju 4/2, onda bi se algoritam odvijao ovako:

**P0000/00=**

**P0000/00=**

**P0000/X0=**

**P0000/X0=**

**P000X/X0=**

**P000X/X0=**

**P000X/XX=**

**P000X/XX=**

**P00XX/XX=**

**P00XX/XX=**

**P00XX/00=**

**P00XX/00=**

**P00XX/X0=**

**P00XX/X0=**

**P0XXX/X0=**

**P0XXX/X0=**

**P0XXX/XX=**

**P0XXX/XX=**

**PXXXX/XX=**

**PXXXX/XX=**

**PXXXX/XX=**

**prihvaća.**

→ ovdje prelazimo u stanje q5, pomičemo glavu udesno  
→ obzirom da nema niti jedne nule, čita se znak =  
→ obzirom da se pročitao znak =, prelazi se u q6, niz se

## **Zaključak**

Konačna verzija ovog algoritma je prvotno prošla kroz 4,5 varijanti, sve do otkrića najjednostavnijeg i najbržeg algoritma. Problemi su bili vezani prvenstveno za način resetiranja broja2, te općenito realizacije algoritma za uzastopno oduzimanje.

Programsko rješenje je priloženo u .rar fajli, a rađeno je upravo po principu Turingovog stroja koji je gore već objašnjen. Bilo bi nespretno objašnjavati kod u samome radu, stoga sam detaljno objašnjenje dao kroz komentare u samom programskom kodu.