

Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva

## **Druga domaća zadaća iz predmeta**

### **“Uvod u teoriju računarstva”**

Zadatak broj 3034

Zagreb, 20. lipanj 2010.

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

**Student:**

**Matični broj studenta:**

**Zadatak broj 3034:** Konstruirati Turingov stroj koji prihvaća sve elemente jezika  $L = \{ww^r w \mid w^r \text{ je obrnuto napisan niz } w\}$ . Niz  $w$  je niz znakova 0 i 1. Turingov stroj implementirati u nekom programskom jeziku i prikazati izvođenje na nekoliko primjera.

# Uvod

Osnovna primjena Turingovog stroja jest prihvaćanje jezika. Pisanje po ulaznoj traci je omogućeno, stoga se Turingov stroj koristi za računanje cjelobrojnih funkcija te za generiranje jezika što je ujedno i moj zadatak.

Osnovni model Turingovog stroja se sastoji od ulazne trake, glave za čitanje i pisanje te upravljačke jedinice. Upravljačka jedinica jest u jednom od konačnog broja stanja. Turingov stroj nakon čitanja znaka ulazne trake zapiše novi znak na traku. 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 sadrži 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 Turingov stroj zapisuje na ulaznu traku čine skup znakova trake.

Tijekom rada upravljačka jedinica donosi odluku na temelju dva podatka:

- stanje;
- znak na traci.

Na temelju pročitanih znakova i stanja jedinice, Turingov stroj odlučuje:

- u koje novo stanje prelazi upravljačka jedinica;
- koji znak se zapisuje na traku umjesto pročitanih znakova;
- u koju stranu se miče glava za čitanje i pisanje.

Turingov stroj formalno se zadaje uređenom sedmorkom:

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

gdje je:

$Q$  – konačan skup stanja;

$\Gamma$  – konačan skup znakova trake;

$B \in \Gamma$  – znak kojim se označava prazna ćelija;

$\Sigma \subseteq (\Gamma - \{B\})$  – konačan skup ulaznih znakova;

$\delta$  – funkcija prijelaza  $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ , gdje je  $L$  i  $R$  označavaju pomak glave u lijevo i desno;

$q_0 \in Q$  – početno stanje;

$F \subseteq Q$  – skup prihvatljivih stanja.

Dozvoljava se da je funkcija prijelaza  $\delta$  nedefinirana za pojedine argumente. Funkcija prijelaza  $\delta(q, V) = (p, Z, W)$  određuje da Turingov stroj 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  $W$  ( $W \in \{L, R\}$ ).

# Ostvarenje

## DEFINICIJA I KONSTRUKCIJA

Turingov stroj za ovaj zadatak se zadaje ovako:

$$ts=(Q, \Sigma, \Gamma, \delta, q\text{-prepis}, B, F)$$

gdje je:

$Q$  – q-prepis, q-vrati-drugu, q-provjera, q-usporedi-kraj, q-usporedi-kraj-vrati-drugu, q-usporedi-kraj-vrati-prvu, q-vrati, q-provjeri-drugu, prihvati;

$\Gamma$  – 0, 1, X, Y, N, J, B;

$B \in \Gamma$  – znak kojim se označava prazna ćelija;

$\Sigma \subseteq (\Gamma - \{B\}) - \{0, 1\}$ ;

$\delta$  – funkcija prijelaza  $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$ , gdje je L i R označavaju pomak glave u lijevo i desno, a N da nema pomaka;

$q\text{-prepis} \in Q$  – početno stanje;

$F \in Q$  – prihvati.

Funkcije prijelaza su ovako definirane:

- 1)  $(q\text{-prepis}, [0, B]) \rightarrow (q\text{-vrati-drugu}, [N, 0], [R, R])$
- 2)  $(q\text{-prepis}, [1, B]) \rightarrow (q\text{-vrati-drugu}, [J, 1], [R, R])$
- 3)  $(q\text{-vrati-drugu}, [0, B]) \rightarrow (q\text{-provjera}, [0, B], [N, L])$
- 4)  $(q\text{-vrati-drugu}, [1, B]) \rightarrow (q\text{-provjera}, [1, B], [N, L])$
- 5)  $(q\text{-provjera}, [1, 1]) \rightarrow (q\text{-provjera}, [Y, J], [R, L])$
- 6)  $(q\text{-provjera}, [0, 0]) \rightarrow (q\text{-provjera}, [X, N], [R, L])$
- 7)  $(q\text{-provjera}, [0, 1]) \rightarrow (q\text{-vrati}, [0, 1], [L, R])$
- 8)  $(q\text{-provjera}, [1, 0]) \rightarrow (q\text{-vrati}, [1, 0], [L, R])$
- 9)  $(q\text{-provjera}, [B, B]) \rightarrow (q\text{-vrati}, [B, B], [L, R])$
- 10)  $(q\text{-provjera}, [0, B]) \rightarrow (q\text{-usporedi-kraj}, [0, B], [N, R])$
- 11)  $(q\text{-provjera}, [1, B]) \rightarrow (q\text{-usporedi-kraj}, [1, B], [N, R])$
- 12)  $(q\text{-vrati}, [X, N]) \rightarrow (q\text{-vrati}, [0, 0], [L, R])$
- 13)  $(q\text{-vrati}, [X, J]) \rightarrow (q\text{-vrati}, [0, 1], [L, R])$
- 14)  $(q\text{-vrati}, [Y, N]) \rightarrow (q\text{-vrati}, [1, 0], [L, R])$
- 15)  $(q\text{-vrati}, [Y, J]) \rightarrow (q\text{-vrati}, [1, 1], [L, R])$
- 16)  $(q\text{-vrati}, [N, B]) \rightarrow (q\text{-prepis}, [N, B], [R, N])$
- 17)  $(q\text{-vrati}, [J, B]) \rightarrow (q\text{-prepis}, [J, B], [R, N])$
- 18)  $(q\text{-vrati}, [B, B]) \rightarrow (q\text{-prepis}, [B, B], [R, N])$

- 19)  $(q\text{-usporedi-kraj}, [1, J]) \rightarrow (q\text{-usporedi-kraj}, [1, 1], [R, R])$
- 20)  $(q\text{-usporedi-kraj}, [0, N]) \rightarrow (q\text{-usporedi-kraj}, [0, 0], [R, R])$
- 21)  $(q\text{-usporedi-kraj}, [B, B]) \rightarrow (q\text{-provjeri-drugu}, [B, B], [N, L])$
- 22)  $(q\text{-usporedi-kraj}, [0, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [0, B], [L, N])$
- 23)  $(q\text{-usporedi-kraj}, [1, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [1, B], [L, N])$
- 24)  $(q\text{-usporedi-kraj}, [1, N]) \rightarrow (q\text{-usporedi-kraj-vrati-drugu}, [1, 0], [N, R])$
- 25)  $(q\text{-usporedi-kraj}, [0, J]) \rightarrow (q\text{-usporedi-kraj-vrati-drugu}, [0, 1], [N, R])$
- 26)  $(q\text{-usporedi-kraj-vrati-prvu}, [0, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [0, B], [L, N])$
- 27)  $(q\text{-usporedi-kraj-vrati-prvu}, [1, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [1, B], [L, N])$
- 28)  $(q\text{-usporedi-kraj-vrati-prvu}, [X, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [0, B], [L, N])$
- 29)  $(q\text{-usporedi-kraj-vrati-prvu}, [Y, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [1, B], [L, N])$
- 30)  $(q\text{-usporedi-kraj-vrati-prvu}, [N, B]) \rightarrow (q\text{-prepis}, [N, B], [R, N])$
- 31)  $(q\text{-usporedi-kraj-vrati-prvu}, [J, B]) \rightarrow (q\text{-prepis}, [J, B], [R, N])$
- 32)  $(q\text{-provjeri-drugu}, [B, 0]) \rightarrow (q\text{-provjeri-drugu}, [B, 0], [N, L])$
- 33)  $(q\text{-provjeri-drugu}, [B, 1]) \rightarrow (q\text{-provjeri-drugu}, [B, 1], [N, L])$
- 34)  $(q\text{-provjeri-drugu}, [B, B]) \rightarrow (q\text{-prihvati}, [B, B], [N, N])$
- 35)  $(q\text{-usporedi-kraj-vrati-drugu}, [0, N]) \rightarrow (q\text{-usporedi-kraj-vrati-drugu}, [0, 0], [N, R])$
- 36)  $(q\text{-usporedi-kraj-vrati-drugu}, [0, J]) \rightarrow (q\text{-usporedi-kraj-vrati-drugu}, [0, 1], [N, R])$
- 37)  $(q\text{-usporedi-kraj-vrati-drugu}, [0, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [0, B], [L, N])$
- 38)  $(q\text{-usporedi-kraj-vrati-drugu}, [1, N]) \rightarrow (q\text{-usporedi-kraj-vrati-drugu}, [1, 0], [N, R])$
- 39)  $(q\text{-usporedi-kraj-vrati-drugu}, [1, J]) \rightarrow (q\text{-usporedi-kraj-vrati-drugu}, [1, 1], [N, R])$
- 40)  $(q\text{-usporedi-kraj-vrati-drugu}, [1, B]) \rightarrow (q\text{-usporedi-kraj-vrati-prvu}, [1, B], [L, N])$

Model Turingovog stroja ima  $k$  ( $k=2$ ) glava za čitanje i pisanje i  $k$  dvostrano beskonačnih traka.

Upravljačka jedinica Turingovog stroja donosi odluku na temelju dviju grupa parametara:

- stanje upravljačke jedinice;
- $k$  pročitanih znakova sa  $k$  traka.

Jednim prijelazom Turingov stroj:

- promijeni stanje;
- zapiše  $k$  znakova na  $k$  traka;
- 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. Druga traka se naziva

radna traka i ona je na početku prazna.

Krenemo sa čitanjem ulaznog niza s lijeva na desno.

Na radnoj traci gradi se podniz koji bi mogao biti naš traženi niz. Zapisuje se znak po znak sa ulazne trake na radnu traku, te se umjesto znaka na ulaznoj traci zapisuje N za nulu odnosno J za jedinicu. Nakon svakog novog zapisanog znaka, provjerava se dobiveni podniz sa radne trake. Druga glava čita podniz s desna na lijevo, dok istovremeno se prva glava kreće s lijeva na desno.

Za svaki idući znak koji odgovara podnizu  $w^r$  stavi se nova oznaka X za nulu odnosno Y za jedinicu na ulaznu traku. Ako naidemo na različite znakove, promijenimo smjer kretanja prve i druge glave, te umjesto oznaka X odnosno Y vraćamo znakove originalnog niza odnosno 0 ili 1. Nakon toga ponavljamo postupak dodavanja novog znaka na kraj podniza, stavljamo oznaku N odnosno J na radnu traku i ponavljamo postupak.

Ako je  $w^r$  odgovarajući obrnuti dio podniza  $w$ , sljedeći je korak provjeriti je li posljednji dio niza jednak podnizu  $w$ . Obje glave pomičemo s lijeva na desno, te za svaki isti znak na radnu traku stavljamo 0 odnosno 1. Ako posljednji dio niza ne odgovara, dodajemo novi znak na kraj podniza radne trake i mijenjamo znak na ulaznoj u njegov odgovarajući 0 ili 1.

Ako je sve do sada bilo ispravno, na kraju posljednjeg koraka imati ćemo na radnoj traci podniz koji tvori  $ww^r w$  na ulaznoj traci. Na kraju treba provjeriti jesmo li ulazni niz pročitali do kraja, ako jesmo, niz se prihvća. Ako nismo, vraćaju se svi znakovi ulazne trake i proširuje se podniz na radnoj traci za još jedan znak.

## **PRIMJER**

Slijedi primjer za ulazni niz 011001.

Prvih pet kolona tablice čini konfiguraciju Turingovog stroja. Konfiguracija Turingovog stroja zadaje se sadržajem ćelija ulazne i radne trake lijevo od glave za čitanje i pisanje, stanjem upravljačke jedinice i sadržajem ćelija ulazne i radne trake koje su desno od glave za čitanje i pisanje.

Sadržaj ul. trake L od glave	Sadržaj rad. trake L od glave	Stanje	Sadržaj ul. trake R od glave	Sadržaj rad. trake R od glave	Funkcija prijelaza
BB...	BB...	q-prepis	011001BB..	BBB....	(q-prepis, [0, B]) -> (q-vrati-drugu, [N, 0], [R, R])
BN	B0	q-vrati-drugu	11001BB..	B0BB...	(q-vrati-drugu, [1, B]) -> (q-provjera, [1, B], [N, L])
BN	B..	q-provjera	11001BB..	0BB...	(q-provjera, [1, 0]) -> (q-vrati, [1, 0], [L, R])
B..	B0	q-vrati	N11001BB..	BB...	(q-vrati, [N, B]) -> (q-prepis, [N, B], [R, N])
BN	B0	q-prepis	11001BB..	BB...	(q-prepis, [1, B]) -> (q-vrati-drugu, [J, 1], [R, R])
BNJ	B01	q-vrati-drugu	1001BB..	BB..	(q-vrati-drugu, [1, B]) -> (q-provjera, [1, B], [N, L])
BNJ	B0	q-provjera	1001BB..	1BB..	(q-provjera, [1, 1]) -> (q-provjera, [Y, J], [R, L])
BNJY	B..	q-provjera	001BB..	0JBB..	(q-provjera, [0, 0]) -> (q-provjera, [X, N], [R, L])
BNJYX	B..	q-provjera	01BB..	BNJBB..	(q-provjera, [0, B]) -> (q-usporedi-kraj, [0, B], [N, R])
BNJYX	B..	q-usporedi-kraj	01BB..	NJBB..	(q-usporedi-kraj, [0, N]) -> (q-usporedi-kraj, [0, 0], [R, R])
BNJYX0	B0	q-usporedi-kraj	1BB..	JBB..	(q-usporedi-kraj, [1, J]) -> (q-usporedi-kraj, [1, 1], [R, R])
BNJYX01	B01	q-usporedi-kraj	BB..	BB...	(q-usporedi-kraj, [B, B]) -> (q-provjeri-drugu, [B, B], [N, L])
BNJYX01	B0	q-provjeri-drugu	BB..	1BB..	(q-provjeri-drugu, [B, 1]) -> (q-provjeri-drugu, [B, 1], [N, L])
BNJYX01	B..	q-provjeri-drugu	BB..	01BB...	(q-provjeri-drugu, [B, 0]) -> (q-provjeri-drugu, [B, 0], [N, L])
BNJYX01	B..	q-provjeri	BB..	B01BB...	(q-provjeri-drugu, [B, B]) -> (prihvati, [B, B], [N, N])
BNJYX01	B..	prihvati	BB..	B01BB...	TS se zaustavlja i prihvata se ulazni niz.

## **PROGRAMSKO RJEŠENJE**

Program se sastoji od dva dijela – razreda *Automat*, te razreda *Pocetna*.

Razred *Pocetna* sadrži main metodu, i pokreće rad *Automat*-a.

Razred *Automat* simulira rad Turingovog stroja. Sastoji se od sljedećih dijelova:

- podrazred *Glava* koji sadrži podrazred *Pomak*
- podrazred *Stanje* koji sadrži podrazred *Prijelaz*

Na početku rada (prilikom instanciranja razreda), *Automat* učitava prijelaze te pozivom metode *pokreni()* započinje simulaciju rada Turingovog stroja. *Glava* simulira rad glave Turingovog stroja. Sastoji se od trake i trenutne pozicije, te sadrži metode za pomicanje glave i čitanje trenutnog znaka. *Pomak* simulira smjer pomaka glave. *Stanje* simulira jedno stanje u kojem se Turingov stroj može nalaziti. Svako stanje sastoji se od jednog ili više *Prijelaz*-a. *Prijelaz* simulira jedan prijelaz Turingovog stroja. *Prijelaz* je jedinstveno definiran ulaznim znakom, izlaznim znakom, pomakom glave, te sljedećim stanjem i sadržan je u *Stanju*.

Za pokretanje Java programa potrebno je imati instalirano Java okruženje (Java JRE). Program se pokreće na način da se pokrene Javina izvršna datoteka s nastavkom *jar*.

Pokretanje datoteke izvršava se u komadnoj liniji i potrebno je upisati sljedeće:

```
java -jar revers.jar imeDatotekeUlazneTrake.txt
```

U datoteci *Primjeri* se nalaze ulazne izlazne datoteke.



## Zaključak

Osnovni model Turingovog stroja moguće je proširiti na više načina, a jedan od tih načina je Turingov stroj s višestrukim trakama. Proširivanje osnovnog modela omogućuje lakše rješavanje zadataka. Rješenje proširenim modelom Turingovog stroja je jednostavno i učinkovito, pa je moguće da je istovjetni osnovni model Turingovog stroja složen. Na primjer, za jedan prijelaz proširenog modela Turingovog stroja, moguće je da istovjetni osnovni model Turingovog stroja izvodi daleko veći broj prijelaza. Za simulaciju jednog pomaka glave Turingov stroj sa  $k$  traka potrebno je mnogo više pomaka glave osnovnog modela Turingovog stroja. Pretpostavi li se da pomak glave troši jedinično vrijeme, smanjivanje više traka Turingovog stroja na jednu traku značajno usporava vrijeme prihvatanja jezika. Ako Turingov stroj ima na raspolaganju dvije trake umjesto jedne, funkcije prijelaza i rad Turingovog stroja se značajno pojednostavljuje.

Zbog gore opisanih razloga, odabrala sam Turingov stroj s dvije trake jer je na taj način puno lakše i prirodnije riješiti zadani problem. Iako dvije trake donose mnoštvo novih kombinacija ulaznih znakova za jedan prijelaz, isto tako donose i veću fleksibilnost Turingovog stroja. Izraditi zadatak nije bilo teško, iako je bilo potrebno dobro razmisliti koji algoritam prepoznavanja niza i obrnutog niza primijeniti.