Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Zvonimir Horvatić

**Druga domaća zadaća iz predmeta**

“**Uvod u teoriju računarstva**”

Zadatak broj **2071**

Zagreb, lipanj 2011.

**Druga domaća zadaća iz predmeta "Uvod u teoriju računarstva"**

**Student**: Zvonimir Horvatić

**Matični broj studenta**: 0036446596

**Zadatak broj 2071**:

|  |
| --- |
| Definirati i programski ostvariti TS koji zadani binarni niz zapisan na traci omeđuje nizom bitova 111 sprijeda i straga te prepravlja zadani niz tako da nakon svakog pojavljivanja niza od dviju binarnih jednica ubacuje binarnu 0 (prepravlja se zadani binarni niz, bez graničnika 111). Npr. za zadani ulazni binarni niz 1011100010 TS kao rezultat daje 11110110100010111. Rad TS-a prikazati po koracima. |

**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.

Osnovni model Turingovog stroja se sastoji od ulazne trake, glave za čitanje i pisanje te

upravljačke jedinke. Upravljačka jedinka 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≥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 jedinka donosi odluku na temelju dva podatka:

1. stanje;
2. znak na traci.

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

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

Turingov stroj formalno se zadaje uređenom sedmorkom:

*ts=(Q, Σ, Γ, δ, q0 , B, F)*

gdje je:

*Q* – konačan skup stanja;

*Γ* – konačan skup znakova trake;

*B Є Γ* – znak kojim se označava prazna ćelija;

*Σ* ⊆ *(Γ-{B})* – konačan skup ulaznih znakova;

*δ* – funkcija prijelaza δ: Q x Γ -> Q x Γ x {L,R} , gdje je L i R označavaju pomak glave u lijevo i desno;

*q0 Є Q* - početno stanje;

*F* ⊆ *Q* – skup prihvatljivih stanja.

Dozvoljava se da je funkcija prijelaza δ nedefinirana za pojedine argumente. Funkcija prijelaza

δ(q, V)=(p, Z, W) odrenuje da Turingov stroj iz stanja q (q Є Q) čitanjem znaka V (V Є Γ) prelazi

u stanje p, na traku zapiše znak Z (Z Є Γ) umjesto znaka V, a glava za čitanje i pisanje miče se u

lijevo ili desno ovisno o W (W Є {L,R}).

**Ostvarenje**

**DEFINICIJA I KONSTRUKCIJA**

Turingov stroj za ovaj zadatak se zadaje ovako:

*ts=(Q, Σ, Γ, δ, q0 , B, F)*

gdje je:

*Q* –{q0, q1, q2, q3, q4, q5, q6, q7, qp};

*Γ* – 0, 1, B;

*B Є Γ* – znak kojim se označava prazna ćelija;

*Σ* ⊆ *(Γ-{B})* – 0, 1;

*δ* – skup funkcija prijelaza δ: Q x Γ -> Q x Γ x {L,R, N} , gdje je L i R označavaju pomak glave u lijevo i desno, a N da nema pomaka;

*q0 Є Q* - početno stanje;

*F* ⊆ *Q* – qp prihvatljivo stanje.

sa sljedećim funkcijama prijelaza:

1. δ(q0, {0, B}) = δ(q1, {0, 1}, {N, R})
2. δ(q1, {0, B}) = δ(q2, {0, 1}, {N, R})
3. δ(q2, {0, B}) = δ(q3, {0, 1}, {N, R})
4. δ(q0, {1, B}) = δ(q1, {1, 1}, {N, R})
5. δ(q1, {1, B}) = δ(q2, {1, 1}, {N, R})
6. δ(q2, {1, B}) = δ(q3, {1, 1}, {N, R})
7. δ(q3, {1, B}) = δ(q4, {1, 1}, {R, R})
8. δ(q3, {0, B}) = δ(q3, {0, 0}, {R, R})
9. δ(q4, {1, B}) = δ(q5, {1, 1}, {N, R})
10. δ(q5, {1, B}) = δ(q3, {1, 0}, {R, R})
11. δ(q4, {0, B}) = δ(q3, {0, 0}, {R, R})
12. δ(q3, {B, B}) = δ(q6, {B, 1}, {N, R})
13. δ(q4, {B, B}) = δ(q6, {B, 1}, {N, R})
14. δ(q6, {B, B}) = δ(q7, {B, 1}, {N, R})
15. δ(q7, {B, B}) = δ(qp, {B, 1}, {N, R})

format zapisa:  
δ (trenStanje, {Traka1, Traka2}) = δ(novoStanje, {Traka1, Traka2}, {Pomak1, Pomak2})

Model Turingovog stroja ima k (k=2) glava za čitanje i pisanje i k traka sa lijevo konačnom ćelijom, a desno beskonačnom trakom.

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.

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

1. stanje upravljačke jedinke;
2. pročitanog znaka sa ulazne trake.

Jednim prijelazom Turingov stroj:

1. promijeni stanje;
2. zapiše znak na radnu traku;
3. pomakne radnu glavu desno dok glavu ulazne trake pomice desno ili ju ne miče

Čitamo niz ulazne trake s lijeva na desno.

Pročitamo prvi znak te neovisno koji je to znak, na radnu traku zapišemo znak 1 te pomaknemo glavu radne trake u desno dok glava ulazne trake ostaje na mjestu.

Ponavljamo tu radnju još dva puta te smo na taj način zapisali lijevi graničnik 111.

Sada čitamo znak po znak ulazne trake te ga prepisujemo na radnu traku pazeći pritom da ukoliko smo pročitali znak 1, prijeđemo u stanje koje "pamti" jedinicu, te ukoliko se pročita 1 još jednom, kaže stroju da u sljedećem prijelazu ne miče glavu ulazne trake, te da na radnu traku zapiše 0. Ovim postupkom nakon svakog 11 znakovlja binarnog niza dodajemo 0.

Kada smo završili s prepisivanjem binarnog niza (ulazna glava pokazuje na praznu ćeliju B), tada krećemo u zadnji korak algoritma, a to je dodavanje desnog graničnika 111 koji se odvija na isti način kao i dodavanje lijevog (ulazna glava stoji, radna se pomiče te se zapisuje znak 1 na radnu traku).

Za kraj samo kažemo stroju da je output konačnog niza radna traka i to je to.

**PRIMJER**

Slijedi primjer za ulazni niz 1011100010.

**BoldUnderline** - oznaka znaka koji se čita / pozicije glave

|  |  |  |  |
| --- | --- | --- | --- |
| Trenutno stanje | Ulazna traka | Radna traka | Funkcija prijelaza |
| q0 | **1**011100010.. | **B**B... | δ(q0, {1, B}) = δ(q1, {1, 1}, {N, R}) |
| q1 | **1**011100010.. | 1**B**B... | δ(q1, {1, B}) = δ(q2, {1, 1}, {N, R}) |
| q2 | **1**011100010.. | 11**B**B... | δ(q2, {1, B}) = δ(q3, {1, 1}, {N, R}) |
| q3 | **1**011100010.. | 111**B**... | δ(q3, {1, B}) = δ(q4, {1, 1}, {R, R}) |
| q4 | 1**0**11100010.. | 1111**B**B... | δ(q4, {0, B}) = δ(q3, {0, 0}, {R, R}) |
| q3 | 10**1**1100010.. | 11110**B**B... | δ(q3, {1, B}) = δ(q4, {1, 1}, {R, R}) |
| q4 | 101**1**100010.. | 111101**B**B... | δ(q4, {1, B}) = δ(q5, {1, 1}, {N, R}) |
| q5 | 101**1**100010.. | 1111011**B**B... | δ(q5, {1, B}) = δ(q3, {1, 0}, {R, R}) |
| q3 | 1011**1**00010.. | 11110110**B**B... | δ(q3, {1, B}) = δ(q4, {1, 1}, {R, R}) |
| q4 | 10111**0**0010.. | 111101101**B**B... | δ(q4, {0, B}) = δ(q3, {0, 0}, {R, R}) |
| q3 | 101110**0**010.. | 1111011010**B**B... | δ(q3, {0, B}) = δ(q3, {0, 0}, {R, R}) |
| q3 | 1011100**0**10.. | 11110110100**B**B... | δ(q3, {0, B}) = δ(q3, {0, 0}, {R, R}) |
| q3 | 10111000**1**0.. | 111101101000**B**B... | δ(q3, {1, B}) = δ(q4, {1, 1}, {R, R}) |
| q4 | 101110001**0**.. | 1111011010001**B**B... | δ(q4, {0, B}) = δ(q3, {0, 0}, {R, R}) |
| q3 | 1011100010**B**.. | 11110110100010**B**B... | δ(q3, {B, B}) = δ(q6, {B, 1}, {N, R}) |
| q6 | 1011100010**B**.. | 111101101000101**B**B... | δ(q6, {B, B}) = δ(q7, {B, 1}, {N, R}) |
| q7 | 1011100010**B**.. | 1111011010001011**B**B... | δ(q7, {B, B}) = δ(qp, {B, 1}, {N, R}) |
| qp | 1011100010**B**.. | 11110110100010111**B**... | Prihvatljivo stanje, Output traka 2 |

1011100010 = **111**011**0**100010**111**

**PROGRAMSKO RJEŠENJE**

Imamo sljedeće datoteke:

1. in.txt (sadrži binarne nizove)
2. start.jar (omogućuje pokretanje programa)
3. folder src gdje se nalaze 3 glavne datoteke programa

**Glavno.java**sadrži kod koji omogućuje čitanje niza iz datoteke in.txt te poziva funkciju TS

**Prijelaz.java** sadrži formalnu definiciju klase prijelaz

**TuringovStroj.java** je srce ovog programa te sadrži sve potrebno za odvijanje programa

Unutar *TuringovStroj.java* postoji nekoliko funkcija koje omogućuju ispravno ponašanje stroja.

npr.

* *NapraviPrijelaze* u kojoj su zadani svi prijelazi;
* *pronadjiPrijelaz* uspoređuje znakove, te traži odgovarajući prijelaz
* *ispisiPrijelaz* koja ispisuje prijelaze

Ukoliko želimo zadati vlastiti primjer na testiranje, dovoljno nam je u datoteku in.txt upisat neki novi binarni niz, uz napomenu da taj niz obavezno moramo zadati u novom retku.

Nakon pokretanja, stvorit će nam se nova tekstualna datoteka formata:

**out -zadanibinarniniz.txt**

Otvorimo li tu datoteku, u njoj ćemo vidjeti ispisane sve prijelaze koji su doveli do stvaranja, tj. preoblikovanja u konačni binarni niz koji je naveden na kraju te iste datoteke.

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 start.jar*** ili pokušati jednostavno dvoklikom na **start.jar**

**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 prihvaćanja jezika. Ako Turingov stroj ima na raspolaganju dvije trake umjesto jedne, funkcije prijelaza i rad Turingovog stroja se značajno pojednostavljuje.

Zbog gore opisanih razloga, odabrao sam Turingov stroj s dvije trake jer je na taj način

puno lakše i prirodnije riješiti zadani problem. Promatramo li složenost konkretnog algoritma, primjetit ćemo da mu je složenost O(n) što je vrlo prihvatljivo, dok bi složenost s jednom trakom bila otprilike O(n\*n).