Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Hrvoje Pozder

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

Zadatak broj 2056

Zagreb, lipanj 2010.

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

**Student:** Hrvoje Pozder

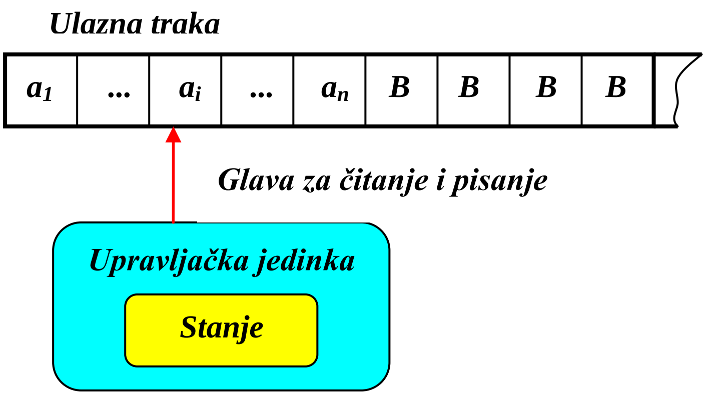
**Matični broj studenta:** 0036446276

**Zadatak broj 2056:** Konstruirati i programski ostvariti Turingov stroj koji prihvaća nizove oblika : $a^i b^{j*i} c^j d^{j-i}$, j>=i>=1.

# Uvod

Turingov stroj (TS) je najopćenitiji poznati matematički model računanja. Bez obzira na njegovu jednostavnost TS ima iste mogućnosti računanja kao bilo koje digitalno računalo. Postoji više modela Turingova stroja a svi se izvode proširivanjem osnovnog modela TS. Osnovni model TS se sastoji od upravljačke jedinice (jedinke), ulazne trake i glave za čitanje i pisanje. Upravljačka se jedinica nalazi u jednom od konačnog broja stanja. Skup stanja TS se sastoji od dva podskupa; skupa prihvatljivih stanja i skupa neprihvatljivih stanja. Ulazna traka TS je podijeljena na ćelije, tj. sastoji se od ćelija koje sadrže znakove trake. Traka ima krajnju lijevu ćeliju, dok je na desnu stranu beskonačna, tj. ima beskonačno mnogo ćelija. Glava za čitanje i pisanje, u skladu sa nazivom, služi za čitanje znakova sa trake i pisanje znakova na traku. TS nakon čitanja znaka s trake mora i zapisati znak na traku. To može biti znak koji je (prethodno) bio pročitan ili neki drugi znak. Glava TS se može pomicati u lijevo ili u desno. Jedina iznimka je slučaj kada glava pokazuje na prvu tj. krajnje lijevu ćeliju trake. Ona se tada ne može pomaknuti u lijevo, već samo u desno. Na početku rada TS $n$ krajnje lijevih ćelija ulazne trake sadrži znakove niza $w$ ( $|w| = n$, $n \geq 0$ ). Ostatak ćelija trake je prazan. Prazne ćelije se označavaju znakom ***B***.

Osnovni model TS je prikazan na *Slici 1*. Znakovi niza $w$ su označeni slovom $a$ i indeksom.




Slika 1: *Osnovni model TS*

Tijekom rada TS njegova upravljačka jedinka, na temelju stanja u kojem se nalazi i znaka koji je pročitan s trake, donosi odluku o tome u koje će novo stanje prijeći, koji će znak biti zapisan na traku na mjesto pročitanog znaka i u koju će se stranu pomaknuti glava za čitanje i pisanje. Donošenje odluke od strane upravljačke jedinke se formalno zapisuje funkcijama prijelaza.

TS se formalno zadaje uređenom sedmorkom:

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

gdje je:

| | |
|---|---|
| Q | - konačan skup stanja; |
| $\Sigma \subseteq (\Gamma - \{B\})$ | - konačan skup ulaznih znakova; |
| $\Gamma$ | - konačan skup znakova trake; |

| | |
|---|---|
| $\delta$ | - 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, respektivno; |
| $q_0 \in Q$ | - početno stanje; |
| $B \in \Gamma$ | - znak kojim se označava prazna ćelija; |
| $F \subseteq Q$ | - skup prihvatljivih stanja. |

$\Gamma$ tj. konačan skup znakova trake se sastoji od znakova ulaznog niza i znakova koje TS zapisuje na traku, ali koji nisu dio ulaznog niza (npr. ***B***). Zato se skup ulaznih znakova definira: $\Sigma \subseteq (\Gamma - \{B\})$, pošto sigurno ne sadrži znak ***B***.

Što se tiče funkcija prijelaza dozvoljeno je da $\delta$ bude nedefinirana za pojedine argumente.

Ako za neko stanje nije definiran odgovarajući prijelaz, rad TS se zaustavlja. Ako je stanje u kojem se TS zaustavio prihvatljivo (tj. ako se nalazi u skupu prihvatljivih stanja) ulazni niz se prihvaća. U suprotnom slučaju (ako stanje nije prihvatljivo) ulazni niz se ne prihvaća.

## Ostvarenje

Potrebno je konstruirati Turingov stroj koji će prihvaćati nizove oblika $a^i b^{j*i} c^j d^{j-i}$, j>=i>=1. Znači broj znakova ***b*** mora biti jednak umnošku broja znakova ***a*** i ***c***, a broj znakova ***d*** mora biti jednak razlici broja znakova ***c*** i ***a***, gdje je minimalan broj znakova ***a*** i ***c*** jednak 1.

Ideja na kojoj se temelji moje rješenje ovog problema je da koristimo Turingov stroj s dvostranom beskonačnom trakom. Ovaj TS se od osnovnog modela razlikuje u tome što nema samo traku beskonačnu u desno, nego i u lijevo. Prilikom konstrukcije ovog TS početno (nulto) polje označit ćemo posebnim znakom **#** koji će nam predstavljati beskonačne prazne ćelije ulijevo. Nakon posebnog znaka slijedi niz znakova ***a,b,c*** i ***d*** koji ako su zadanog oblika se prihvaćaju, a ako nisu zadanog oblika onda se ne prihvaćaju.

Najmanji mogući niz je kad su *j=i=1*, i izgleda ***abc***. Ovo je ujedno i poseban slučaj gdje je broj znakova ***a*** i ***c*** jednak, pa se u nizu ne može pojaviti znak ***d*** (*j-i=0*).

Npr. neka imamo zadan niz ***aabbbbbbcccd***. Ovaj niz jest zadanog oblika i on je prihvatljiv. Ideja na koji način će moj Turingov stroj ispitati je li on prihvatljiv je sljedeća :

1. u početnom stanju *q0* ulazni znak mora biti ***a*** (*min. i=1*), te ga prepiši s ***A***, prijeđi u novo stanje *q1* i 'šeći' u desno dok ne nađeš prvi ***c*** (*min. j=1*).

2. kad nađeš prvi ***c*** prepiši ga s ***C***, prijeđi u novo stanje *q2* i traži prvi lijevi ***b***.

3. kreći se lijevo-desno i za svaki ***c*** prepiši jedan ***b*** sa ***B***, tako sam ostvario množenje prvog ***a*** sa svim znakovima ***c***.

4. kad više ne nađe ni jedan ***c***, doći će do ***d*** ili ***epsilon*** (prazna ćelija u slučaju ako su *j=i*), prijeđi u novo stanje, vrati se do prvog lijevog ***a***, te po putu prepiši sve ***C*** sa ***c***.

5. ponavljaj postupak dok ne izmnožiš sve ***a*** sa ***c***, time su svi ***b*** prepisani sa ***B***, ***a*** sa ***A*** i ***c*** sa ***C***.

6. pošto je izmnožio i provjerio odgovara li broj b-ova, više ne može naći ***a***, prijeđi u novo stanje, idi na početak niza i po putu vrati sve ***A*** i ***C*** u ***a*** i ***c***.

7. za svaki ***a*** prepisan sa ***A***, prepiši jedan ***c*** sa ***C***.

8. kad više ne možeš naći ***a*** (znači oduzeti su svi), na traci je ostao ostatak znakova ***c*** koji mora odgovarati broju znakova ***d*** (pošto je ***d*** ostatak oduzimanja ***c-a***), pa za svaki preostali znak ***c*** prepiši jedan znak ***d*** sa ***D***. U slučaju kad je *j=i* onda ne smije biti znakova ***d*** na traci.

9. Ako su svi znakovi na traci uspješno prepisani s odgovarajućim velikim slovima i glava je došla do kraja niza, niz je prihvatljiv.

Pošto je ovo dosta zbunjujuće i zvuči komplicirano, najbolje da ovaj primjer ilustriram grafički.

Moj model imat će sljedeću defniciju :

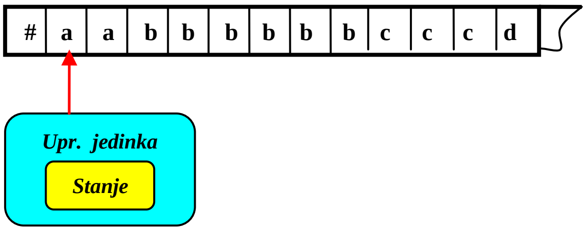
Skup stanja : (q0,q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q12,qf)

Abeceda trake : (a,b,c,d,A,B,C,D,#,E)

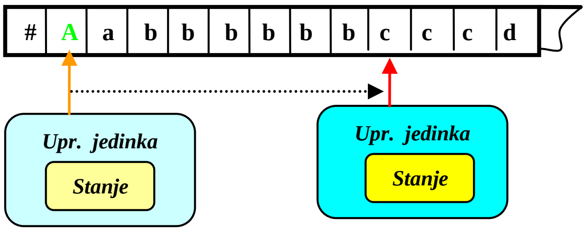
Početno stanje : (q0)

Prihvatljiva stanja : (qf)

**NAPOMENA** : U mojem modelu sam prazne ćelije označavao s ***E***, jer sam ***B*** koristio u druge namjene

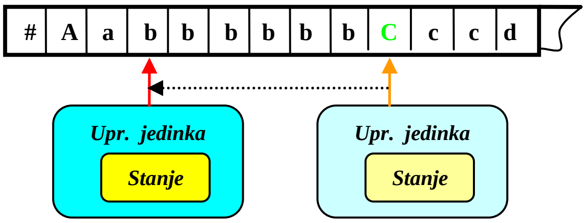



Slika 2 : *Početno stanje*

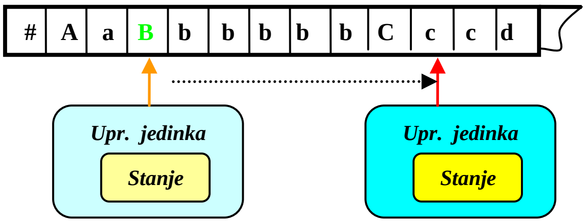



Slika 3 : (q0,a)=(q1,A,R)
(q1,a)=(q1,a,R)
(q1,b)=(q1,b,R)

Na početku na traci se nalazi niz ***aabbbbbbcccd***, a lijevo i desno od njega beskonačno mnogo praznih ćelija. U mojem slučaju beskonačne ćelije u lijevu stranu označene su posebnim znakom *#*. Glava za čitanje se u početku nalazi na prvom lijevom znaku ***a***, a Turingov stroj je u početnom stanju *q0*. Za ulazni znak ***a*** (kojeg prepisuje znakom ***A***) TS prelazi u stanje *q1* koje mi služi za šetnju udesno preko svih znakova do prvog ***c***, kako bi izračunali množenje a*c.

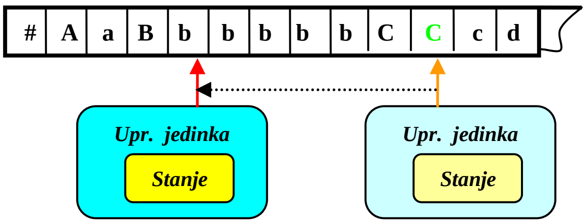



Slika 4 : (q1,c)=(q2,C,L)
(q2,b)=(q2,b,L)
(q2,a)=(q3,a,R)




Slika 5 : (q3,b)=(q1,B,R)
(q1,b)=(q1,b,R)
(q1,C)=(q1,C,R)

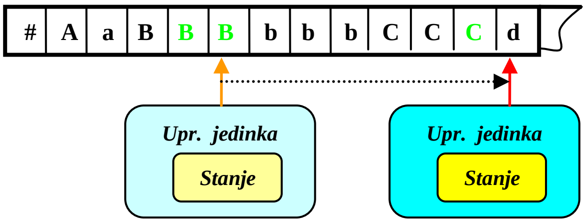
Kad stigne do ***c*** (kojeg prepisuje znakom ***C***) TS prelazi u stanje *q2* koje služi za šetnju ulijevo preko svih znakova do prvog lijevog znaka ***b***. Tada njega prepisuje sa ***B***, te prelazi ponovno u stanje *q1*.



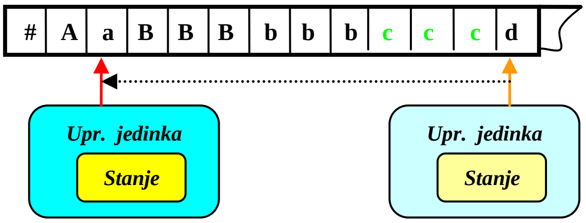

Slika 6 : (q1,c)=(q2,C,L)
(q2,C)=(q2,C,L)
(q2,b)=(q2,b,L)
(q2,B)=(q3,B,R)

**•••**

TS naizmjenično za svaki ***c*** kojeg prepiše sa ***C***, prepisuje po jedan ***b*** sa ***B***, i time sam ostvario množenje jednog ***a*** sa svim znakovima ***c***.

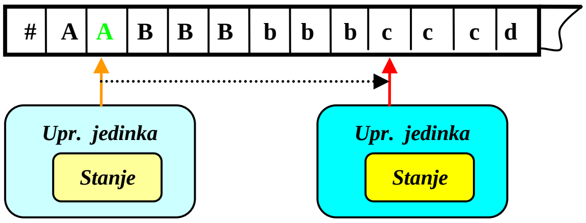



Slika 7 : (q3,b)=(q1,B,R)
(q1,b)=(q1,b,R)
(q1,C)=(q1,C,R)

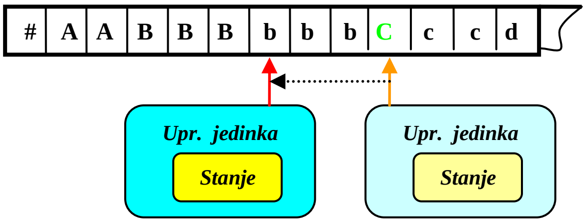



Slika 8 : (q1,d)=(q4,d,L)
(q4,C)=(q4,c,L)
(q4,b)=(q4,b,L)
(q4,B)=(q4,B,L)
(q4,a)=(q4,a,L)
(q4,A)=(q0,A,R)

Kad se svi znakovi ***c*** prepišu sa ***C***, TS će u stanju *q1* doći u ovom slučaju do znaka ***d*** (u slučaju kad bi bilo *j=i* onda ne bi bilo znakova ***d*** i glava bi došla do prazne ćelije gdje bi umjesto prijelaza (q1,d)=(q4,d,L) iskoristila prijelaz (q1,E)=(q4,E,L)). Tad TS prelazi u stanje *q4* koje služi za šetnju ulijevo do prvog lijevog znaka ***a*** kako bi započela nova iteracija množenja, i po putu resetira (odnosno prepisuje) sve znakove ***C*** u znakove ***c***.

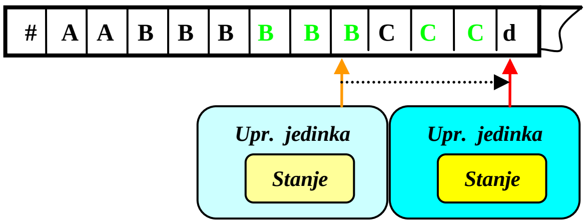



Slika 9 : (q0,a)=(q1,A,R)
(q1,B)=(q1,B,R)
(q1,b)=(q1,b,R)




Slika 10 : (q1,c)=(q2,C,L)
(q2,b)=(q2,b,L)
(q2,B)=(q3,B,R)

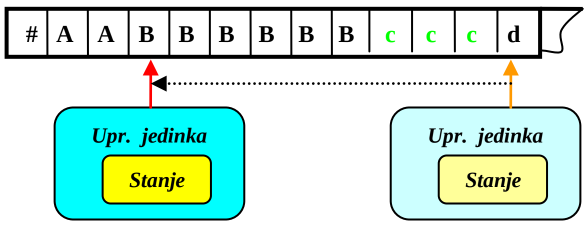
Kad TS prijeđe iz stanja *q4* u stanje *q0* postupak se ponovno ponavlja, te se obavlja druga (ujedno i završna) iteracija množenja a*c.



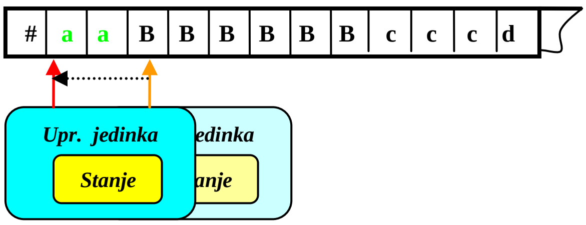

•••

Slika 11 : (q3,b)=(q1,B,R)
(q1,C)=(q1,C,R)

Sada nakon množenja a*c svi znakovi ***b*** (ako je niz dobro zadan) bi trebali biti prepisani sa ***B***.

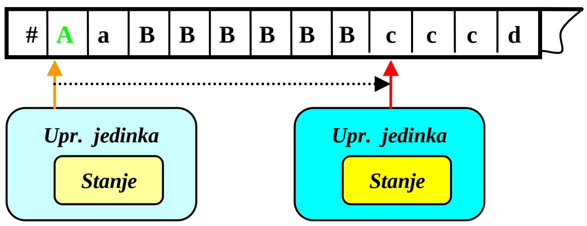



Slika 12 : (q1,d)=(q4,d,L)
(q4,C)=(q4,c,L)
(q4,B)=(q4,B,L)
(q4,A)=(q0,A,R)

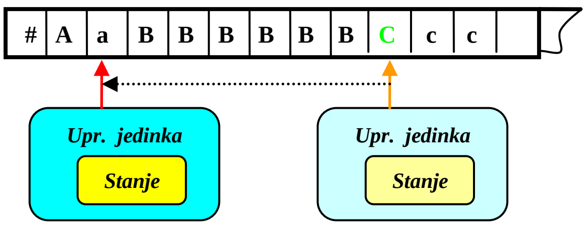



Slika 13 : (q0,B)=(q5,B,L)
(q5,A)=(q5,a,L)
(q5,#)=(q6,#,R)

TS u stanju *q1* i ulazni znak ***d*** prelazi u stanje *q4* u kojem se opet vraća ulijevo do prve ćelije desno od zadnjeg znaka ***A***, te po putu resetira (odnosno prepisuje) sve znakove ***C*** sa ***c***. Prelazimo u stanje *q0* kojem više ulazni znak nije ***a*** nego ***B***, što znači da se množenje obavilo u potpunosti te možemo prijeći na oduzimanje. Tad TS prelazi u stanje *q5* kojim se krećemo ulijevo do graničnika **#**, te po putu resetiramo sve znakove ***A*** u ***a***.

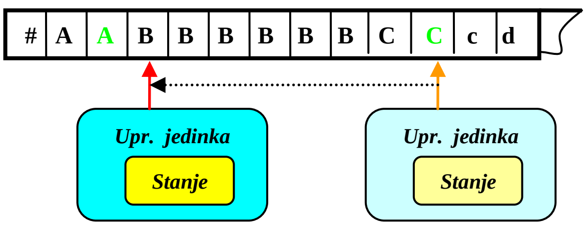



Slika 14 : (q6,a)=(q7,A,R)
(q7,a)=(q7,a,R)
(q7,B)=(q7,B,R)




Slika 15 : (q7,c)=(q8,C,L)
(q8,B)=(q8,B,L)
(q8,a)=(q8,a,L)
(q8,A)=(q6,A,R)

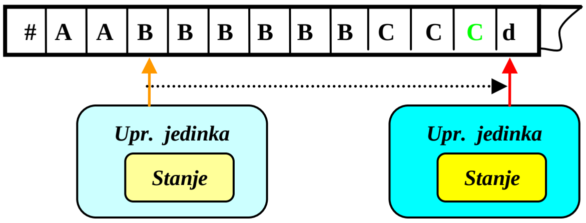
U stanju *q6* prepisujemo prvi znak ***a*** sa ***A*** te prelazimo u stanje *q7* koje nam služi za šetnju udesno sve do prvog znaka ***c***. Kad dođemo do ***c*** (kojeg prepisujemo sa ***C***), prelazimo u stanje *q8* koje služi za šetnju ulijevo preko svih znakova tražeći prvi lijevi znak ***a***.



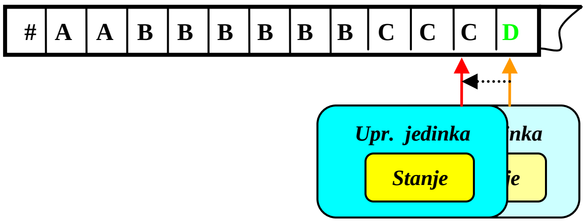

•••

Slika 16 : (q7,c)=(q8,C,L)
(q8,C)=(q8,C,L)
(q8,B)=(q8,B,L)
(q8,A)=(q6,A,R)

Sada naizmjenično za svaki znak ***a*** prepisan sa ***A*** prepišemo po jedan znak ***c*** sa ***C***. Time sam ostvario oduzimanje.

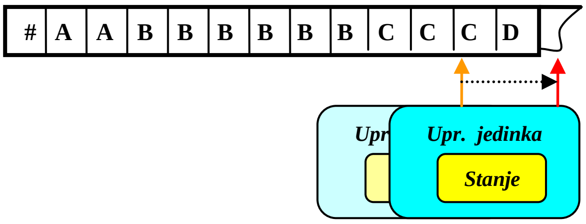



Slika 17 : (q6,B)=(q9,B,R)
(q9,B)=(q9,B,R)
(q9,C)=(q9,C,R)
(q9,c)=(q10,C,R)




Slika 18 : (q10,d)=(q11,D,L)

Kad u stanju *q6* ulazni znak više nije znak ***a*** nego znak ***B*** to znači da smo sve znakove ***a*** oduzeli od znakova ***c***, te je ostao ostatak znakova ***c*** koji je jednak ostatku oduzimanja c-a, koji ujedno mora biti jednak (ako je niz dobro zadan) broju znakova ***d***. Prelazimo u stanje *q9* kako bi išli na taj posljednji korak. U tom stanju prelazimo preko svih znakova tražeći prvi znak ***c***. Ne mora biti nužno da ćemo ga naći (u slučaju *j=i* razlika je 0, te nema ostatka ni znakova ***d***), pa ako dođemo do kraja niza prelazimo odmah u prihvatljivo stanje prijelazom (q9,E)=(qf,E,L). Ako naiđemo na znak ***c***, prepisujemo ga sa ***C*** i prelazimo u stanje *q10* koje služi za šetnju udesno (u slučaju više znakova ***c***) do prvog znaka ***d***. Zatim znak ***d*** prepisujemo sa znakom ***D*** i prelazimo u stanje *q11* koje služi za šetnju ulijevo.




Niz je prihvatljiv!

Slika 19 : (q11,C)=(q9,C,R)
(q9,D)=(q12,D,R)
(q12,E)=(qf,E,L)

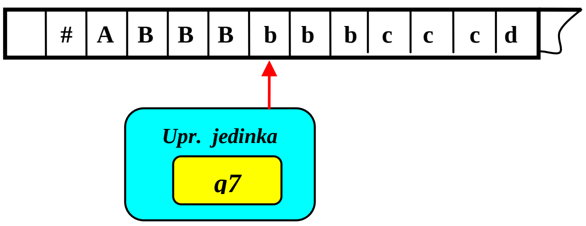
Kad u stanju *q9* ulazni znak više nije ***c*** nego ***D***, to znači da je završena u potpunosti razlika c-a i da su svi znakovi početnog niza prepisani s odgovarajućim velikim znakovima. Tad prelazimo u stanje *q12* koje služi da se prošećemo do kraja niza (tj. da obavimo provjeru da ne bi bilo viška znakova ***d***), te prelazimo u stanje *qf* koje je ujedno i jedino prihvatljivo stanje. TS donosi odluku je li niz prihvatljiv tek kad se zaustavi. Stroj se zaustavlja kad za ulazni znak i stanje u kojem se nalazi nema definiran prijelaz. Pošto u našem slučaju stanje *qf* nema definiranih prijelaza TS se zaustavlja. Sada TS može donijeti jednu od dvije odluke : niz je prihvatljiv, ili niz nije prihvatljiv. To ovisi je li stanje u kojem se TS prilikom zaustavljanja nalazi u skupu prihvatljivih stanja ili ne. Stanje *qf* se nalazi u skupu prihvatljivih stanja, pa se naš niz prihvaća. Da se stroj nalazio u bilo kojem drugom stanju prilikom zaustavljanja, niz ne bi bio prihvatljiv.

Tablica prijelaza :

| | a | b | c | d | A | B | C | D | # | E |
|---|---|---|---|---|---|---|---|---|---|---|
| q0 | q1,A,R | - | - | - | - | q5,B,L | - | - | - | - |
| q1 | q5,B,L | q1,b,R | q2,C,L | q4,d,L | - | q1,B,R | q1,C,R | - | - | q4,E,L |
| q2 | q3,a,R | q2,b,L | - | - | q3,A,R | q3,B,R | q2,C,L | - | - | - |
| q3 | - | q1,B,R | - | - | - | - | - | - | - | - |
| q4 | q4,a,L | q4,b,L | - | - | q0,A,R | q4,B,L | q4,c,L | - | - | - |
| q5 | - | - | - | - | q5,a,L | - | - | - | q6,#,R | - |
| q6 | q7,A,R | - | - | - | - | q9,B,R | - | - | - | - |
| q7 | q7,a,R | - | q8,C,L | - | - | q7,B,R | q7,C,R | - | - | - |
| q8 | q8,a,L | - | - | - | q6,A,R | q8,B,L | q8,C,L | - | - | - |
| q9 | - | - | q10,C,R | - | - | q9,B,R | q9,C,R | q12,D,R | - | qf,E,L |
| q10 | - | - | q10,c,R | q11,D,L | - | - | - | q10,D,R | - | - |
| q11 | - | - | q11,c,L | - | - | - | q9,C,R | q11,D,L | - | - |
| q12 | - | - | - | - | - | - | - | q12,D,R | - | qf,E,L |
| qf | - | - | - | - | - | - | - | - | - | - |

Velika vjerojatnost je da ulazni niz bude pogrešno zadan, tj. da bude neprihvatljiv. Jedna mogućnost će se desiti ako bude viška ili manjka nekih od znakova ***a***, ***b***, ***c*** ili ***d***.

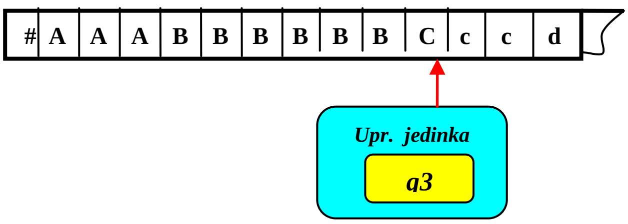
1. Na primjer, da je manjak znakova ***a***, tada bi se nalazili u ovoj situaciji :




Slika 20

Situacija je slična situaciji na *Slici 14.* TS je u stanju *q7* i za svaki znak ***a*** prepisan sa ***A*** treba prepisati po jedan znak ***c*** sa ***C***. No kako postoje još uvijek znakovi ***b*** (koji su prisutni jer broj znakova ***b*** ne odgovara umnošku a*c) i nije definiran prijelaz za stanje *q7* i ulazni znak ***b***, TS prilikom šetanja udesno staje i niz se ne prihvaća.

2. Za višak znakova ***a*** imamo sljedeći slučaj :




Slika 21

Sada se nalazimo u situaciji sličnoj situaciji na *Slici 10*. Treći znak ***a*** smo prepisali sa ***A*** i krenuli smo u treću iteraciju množenja a*c, tj za svaki znak ***c*** prepisan sa ***C***, prepisati po jedan znak ***b*** sa ***B***. Pošto mi nemamo više znaka ***b***, a za stanje *q3* i ulazni znak ***C*** nije definiran prijelaz, TS staje i niz se ne prihvaća.

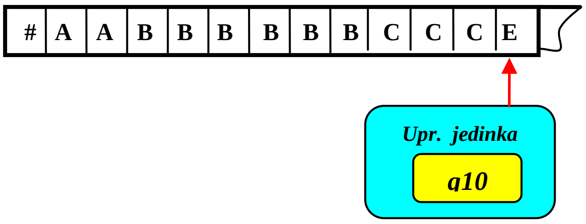
3. Slučaj kad je manjak znakova ***b*** je identičan prethodnom slučaju kad je višak znakova ***a***.

4. Slučaj kad je višak znakova ***b*** opet se svodi na 1. slučaj, odnosno neće se uspjeti svi znakovi ***b*** prepisati znakovima ***B*** pošto njihov broj ne odgovara broju a*c, te u situaciji kao na *Slici 20* za stanje *q7* i ulazni znak ***b*** nije definiran prijelaz, pa TS staje i niz se ne prihvaća.

5. Slučaj kad je manjak znakova ***c*** svodi se na slučajeve 1. i 4.

6. Slučaj kad je višak znakova ***c*** svodi se na slučajeve 2. i 3., odnosno javit će se manjak znakova ***b***. Za stanje *q3* i ulazni znak ***C*** nije definiran prijelaz, pa TS staje i niz se ne prihvaća.

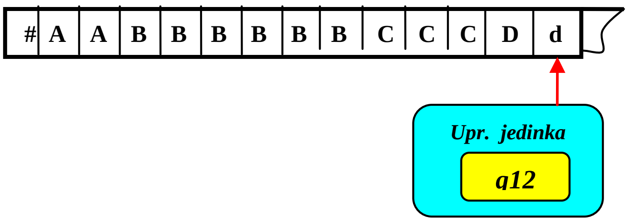
7. Za manjak znakova ***d*** imamo sljedeći slučaj :




Slika 22

Nalazimo se u situaciji sličnoj kao na *Slici 17*. Nakon što je izvršeno oduzimanje TS prelazi u stanje *q9* i prelazi preko svih znakova do ostatka znakova ***c***. Tad prvi znak ***c*** prepisuje sa ***C***, prelazi u stanje *q10* koje traži prvi lijevi znak ***d*** da ga prepiše sa ***D***, no pošto nedostaje znakova ***d***, TS je došao do kraja niza, i kako za stanje *q10* i ulazni znak ***E*** (prazna ćelija) nije definiran prijelaz, TS staje i niz se ne prihvaća.

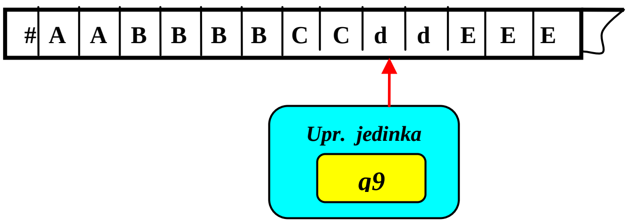
8. Za višak znakova ***d*** imamo sljedeći slučaj :




Slika 23

Nalazimo se u situaciji sličnoj kao na *Slici 19*. TS je za svaki ostatak slova ***c*** prepisao po jedan znak ***d*** znakom ***D***. Nakon toga je prešao u stanje *q12* kojim se trebao prošetati do kraja niza i upravo provjeriti da ne bi bilo viška znakova ***d***. Pošto ih ima, a za stanje *q12* i ulazni znak ***d*** ne postoji prijelaz, TS staje i niz se ne prihvaća.

9. Ako je *j=i*, odnosno broj znakova ***a*** jednak broju znakova ***c***, te postoje znakovi ***d*** koji ne bi trebali postojati (*j-i=0*), imamo sljedeći slučaj :




Slika 24

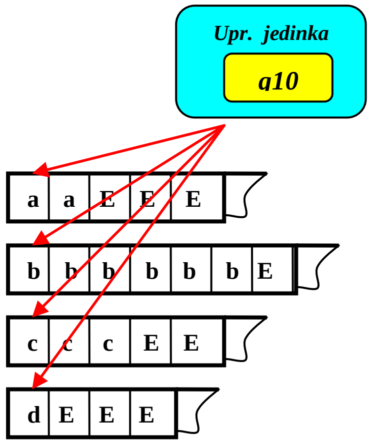
Nalazimo se u situaciji sličnoj kao na *Slici 17*. TS je izvršio oduzimanje c-a i u stanju *q9* traži ostatak znakova ***c*** ili kraj niza u slučaju kad je *j=i*. Pošto mi imamo zabranjene znakove ***d***, za stanje *q9* i ulazni znak ***d*** nije definiran prijelaz, pa se TS zaustavlja i niz se ne prihvaća.

Druga mogućnost krivog unosa će se desiti ako se poremeti redoslijed znakova, odnosno ako se neće pojaviti znakovi ***a***, ***b***, ***c*** i ***d*** respektivno u bilo kojoj količini. Tada niz također neće biti prihvatljiv, jer će se također doći do situacije kada neće postojati prijelaz prije nego TS dođe u prihvatljivo stanje.

# Zaključak

Turingov stroj je moćan alat kojim možemo realizirati teške probleme. Kad sam prvi put pročitao svoj zadatak, zvučao mi je dosta komplicirano te sam znao da neću lako doći do rješenja problema. No razmišljajući kao Turingov stroj i za neki zadani ulazni niz brzo su počele stizati ideje.

Prva ideja je bila realizirati problem pomoću Turingovog stroja sa višestrukim trakama. Tako bi ulazni niz razdijelio po znakovima ***a***, ***b***, ***c*** i ***d*** u četiri dijela, te svaki niz znakova stavio u zasebnu traku. Tada bi početno stanje Turingovog stroja za niz ***aabbbbbbcccd*** izgledalo ovako :




*Slika 25* : Turingov stroj s višestrukim trakama

Tu se javio problem što kod svakog prijelaza meni nije odgovaralo da se pomaknu sve glave Turingovog stroja, već samo neke što se kosilo sa definicijom Turingovog stroja. Međutim analiziranjem se ispostavilo da ako i glava ostane na mjestu, Turingov stroj će ostati ekvivalentan osnovnom modelu, što je pak trebali dokazati. Ispalo je da je ovo rješenje ispravno ali dosta komplicirano, pa sam odlučio potražiti drugo jednostavnije rješenje. Drugo rješenje je sa dvostranom beskonačnom trakom, kojim sam u konačnici i ostvario svoj model Turingovog stroja za zadani problem.

Sljedeći problem koji se javio je bio kako kod programskog ostvarenja osigurati dovoljno veliko polje za proizvoljan ulazni niz. Međutim, pošto je program pisan u programskom jeziku *perl*, lako se otklonio ovaj problem. Naime, polje u perlu je po *defaultu* beskonačno te se za svaki znak upisan iza granice polja, polje automatski proširuje. Jedino ograničenje je memorijski prostor računala.

Posljednji od mogućih problema je bilo obuhvaćanje svih kombinacija ulaznih nizova i posebnih slučajeva. Detaljnim analiziranjem došao sam do zaključka da vrijedi nekoliko posebnih slučajeva :

- ulazni niz je minimalan (za $j=i=1$ niz izgleda ***abc***)
- ulazni niz ima jednak broj znakova ***a*** i znakova ***c*** (za $j=i>1$ niz izgleda npr. ***aabbbbcc***)
- ulazni niz ima različit broj znakova ***a*** i znakova ***c*** (za $j>i>=1$ niz izgleda ***abbccd***)

Ostalo je još samo staviti zaštite od pogrešno zadanih nizova.

Nakon rješavanja ovih problema dalje je sve išlo glatko, te sam uspješno konstruirao i programski ostvario Turingov stroj za zadani problem.