

Sveučilište u Zagrebu
Fakultet Elektrotehnike i Računarstva

marac88

(kolega Ivok, fala ti puno!!)

Druga domaća zadaća iz kolegija
"Uvod u teoriju računarstva"

Zadatak broj 2069.

Zagreb, lipanj, 2009. godine

Druga domaća zadaća iz kolegija "Uvod u teoriju računarstva"

student: marac88

matični broj studenta: ***

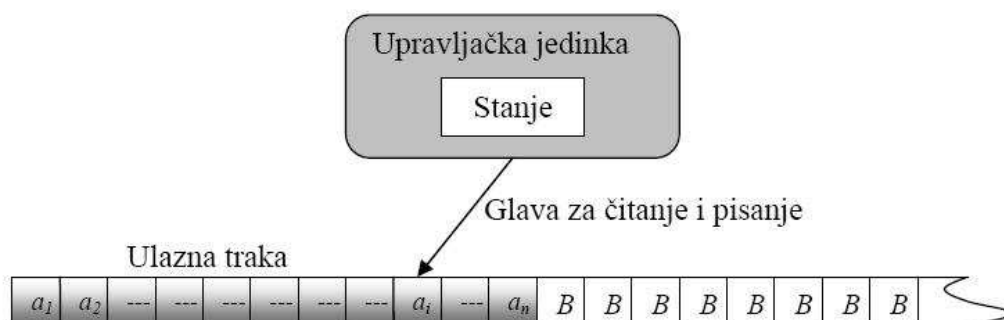
zadatak broj 2069.: Konstruirati i programski simulirati Turingov stroj koji prebrojava pojavu određenog uzorka u tekstu. Početni sadržaj trake je `uzorak#tekst`, a na kraju rada Turingov stroj na proizvoljnom mjestu na traci mora ispisati broj pojavljivanja uzorka.

1. Uvod

Turingov stroj (TS) je najopćenitiji poznati matematički model računanja. Postoji više modela Turingova stroja a svi se izводе proširivanjem osnovnog modela TS.

Turingov stroj se sastoji od upravljačke jedinice koja uz pomoć glave za čitanje i pisanje, čita i zapisuje znakove sa i na ulaznu traku. Upravljačka se jedinica nalazi u jednom od konačnog broja stanja koja mogu biti prihvatljiva i neprihvatljiva. Ulazna traka nije beskonačna sa obje strane, već je sa lijeve strane ograničena, a sa desne strane neograničena i sastoji se od ćelija koje sadrže znakove trake. Glava za čitanje i pisanje, u skladu sa nazivom, služi za čitanje znakova sa trake i pisanje znakova na traku te se može pomicati u lijevo i u desno. Iznimka je kada glava pokazuje na krajnje lijevu ćeliju i tada se može pomaknuti samo u desno. Nakon čitanja znaka s trake, stroj mora i zapisati novi znak koji je ili već bio pročitani ili neki novi. Na početku rada n krajnje lijevih ćelija sadrže niz w gdje je $|w| = n$, a $n \geq 0$, znakovi tog niza i znakovi koje Turingov stroj zapisuje na traku čine skup znakova trake. Ostatak trake je popunjen praznim ćelijama koje se označavaju sa B. Znakovi niza w su označeni slovom a i indeksom. Tijekom rada na temelju pročitanih znakova i trenutnog stanja upravljačke jedinice, Turingov stroj odlučuje: u koje novo stanje jedinica prelazi, koji znak se zapisuje na traku umjesto pročitanih znakova i u koju se stranu pomiče glava za čitanje i pisanje. Donošenje odluke od strane upravljačke jedinice se formalno zapisuje funkcijama prijelaza.

Osnovni model Turingovog stroja je prikazan na slici 1.



Slika 1: Osnovni model TS

Turingov stroj se zadaje kao uređena sedmorka:

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

gdje je

Q - konačan skup stanja

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

Γ - konačan skup znakova trake

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

$B \in \Gamma$ - znak kojim se označava prazna ćelija

$F \subseteq Q$ - skup prihvatljivih stanja

Također, dozvoljava se da je $\delta(q, V) = (p, Z, W)$ nedefinirana za pojedine argumente te da određuje da stroj iz stanja q ($q \in Q$) čitanjem znaka V ($V \in \Gamma$) prelazi u stanje p , dok se na traku zapisuje znak Z ($Z \in \Gamma$) umjesto znaka V , a glava za čitanje se pomiče lijevo ili desno ovisno o W ($W \in \{L, R\}$).

2. Ostvarenje

Cilj ovog zadatka je da prebroji pojavu uzorka u tekstu koristeći princip Turingova stroja. Dakle, Turingov stroj će imati potrebna stanja, znakove zapisane na traci i tablicu prijelaza.

Za potrebe zadatka, uzorak i tekst se mogu sastojati od maksimalno 8 znakova, tj. od prvih 8 slova abecede (a, b, c, d, e, f, g, h). Naravno, mogla se uzeti i cijela abeceda, no tada bi tablica prijelaza bila puno kompleksnija i dulja, a princip bi ostao isti.

Formalno zadavanje Turingovog stroja čija se izgradnja traži u tekstu zadatka je:

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

gdje je:

- $Q = \{q_0, q_1a, q_1b, q_1c, q_1d, q_1e, q_1f, q_1g, q_1h, q_2a, q_2b, q_2c, q_2d, q_2e, q_2f, q_2g, q_2h, q_3a, q_3b, q_3c, q_3d, q_3e, q_3f, q_3g, q_3h, q_4a, q_4b, q_4c, q_4d, q_4e, q_4f, q_4g, q_4h, q_{11a}, q_{11b}, q_{11c}, q_{11d}, q_{11e}, q_{11f}, q_{11g}, q_{11h}, q_{22a}, q_{22b}, q_{22c}, q_{22d}, q_{22e}, q_{22f}, q_{22g}, q_{22h}, q_5, q_e, (p+1), q_{krivo}, q_{krivo1}, q_{krivo2}, q_{krivo3}\}$
- $\Sigma = \{a, b, c, d, e, f, g, h, \#\}$
- $\Gamma = \{a, b, c, d, e, f, g, h, \#, A, *, C, D, E, F, G, H, B\}$
- $B \in \Gamma$ = znak kojim se označava prazna ćelija
- $q_0 \in Q$ = početno stanje
- δ = funkcija prijelaza, D označava desno kretanje glave, a L lijevo
- $F = q_e$

Programsko rješenje je isprogramirano u C-u i gotovo cijelo rješenje (uz par preinaka) je preuzeto sa 3. laboratorijske vježbe i simuliranja rada Turingova stroja.

Princip kako je riješen zadatak je takav da postoji jedna traka na kojoj je s jedne strane # upisan uzorak, a s druge strane tekst u kojem se mora taj uzorak pronaći. I uzorak i tekst su zapisani malim slovima što je omogućilo da se u tablici prijelaza lakše "pamte" stanja te da ih ima (relativno) malo u odnosu na to koliko bi ih bilo da se zapisuje i malim i velikim slovima. Cijelo rješenje radi na tom principu da glava za čitanje i pisanje se kreće od početka uzorka,

označuje si pročitano slovo (malo slovo pretvara u veliko), kreće se dalje do # i prelazi u tekst. Kada je glava za čitanje prešla u tekst, traži to pročitano slovo na način da slova koja nisu prihvatljiva pretvori u nule i traži dalje. U trenutku kada konačno nađe traženo slovo, opet ga označuje te odlazi natrag u uzorak provjeriti koje je sljedeće slovo po redu kako bi mogla dalje tražiti. U slučaju da u uzorku ne postoje nova slova, glava izbacuje jedinicu na krajnje lijevi dio trake, prije početka samog uzorka, dok se u slučaju da u uzorku postoje nova slova cijeli proces ponavlja. Broj jedinica koji se ispisuju na krajnje lijevom dijelu trake označava broj pojavljivanja uzorka u cjelokupnom tekstu (npr. 111 = 3 pojavljivanja).

2.1 Tablica prijelaza

Uz programsku dokumentaciju, priložena je i tablica prijelaza ovog Turingovog stroja. No, s obzirom na veliki broj kombinacija (preko 700), objasniti ću ovdje samo segmente tablice prijelaza.

Zbog lakšeg objašnjenja, uvesti ću nove oznake:

- x_1 = jedno malo slovo iz skupa svih malih slova (a, b, c, d, e, f, g, h)
- x_n = bilo koje slovo iz skupa svih malih slova (a, b, c, d, e, f, g, h)
- Y_1 = jedno veliko slovo iz skupa svih velikih slova (A, B, C, D, E, F, G, H)
- Y_n = bilo koje slovo iz skupa svih velikih slova (A, B, C, D, E, F, G, H)
-

2.1.1 Segmenti tablice prijelaza

- $\delta(q_0, x_1) = (q_1x_1, Y_1, D)$
 - glava počinje čitati od krajnje lijevog znaka uzorka i "predaje" taj znak stanju q_1x_1 i označava ga (veliko slovo)
- $\delta(q_0, 1) = (q_0, 1, D)$
 - glava preskače jedinice (broj pojavljivanja uzorka) prilikom "povratka" na početak uzorka

- $\delta (q_1x_1, x_n) = (q_1x_1, x_n, D)$
 - glava preskače dio uzorka, nakon prvog pročitano­g znaka uzorka, do #
- $\delta (q_1x_1, \#) = (q_2x_1, \#, D)$
 - nakon što glava pročita #, prelazi u stanje q_2x_1
- $\delta (q_2x_1, x_1) = (q_3x_1, Y_1, L)$
 - glava traži slovo i onda ga označava (veliko slovo)
- $\delta (q_2x_1, (x_n-x_1+0)) = (q_2x_1, 0, D)$
 - glava mijenja sva slova, osim " x_1 ", u 0 i pomiče se desno dok ne dođe do prazne ćelije ("B")
- $\delta (q_2x_1, B) = (q_e, B, L)$
 - glava je prošla cijeli tekst
- $\delta (q_2x_1, Y_1) = (q_2x_1, 0, D)$
 - glava prebacuje sva velika slova Y_1 u 0
- $\delta (q_3x_1, (Y_1+0)) = (q_3x_1, (Y_1+0), L)$
 - glava na "povratku" preskače sve znakove do #
- $\delta (q_3x_1, \#) = (q_4x_1, \#, L)$
 - nakon što je glava sve preskočila, dolazi do # i prelazi u stanje q_4x_1
- $\delta (q_4x_1, x_1) = (q_4x_1, x_1, L)$
 - glava se vraća ulijevo od # i preskače sva mala slova
- $\delta (q_4x_1, Y_1) = (q_5, Y_1, D)$
 - glava je našla veliko slovo i prelazi u novo stanje q_5
- $\delta (q_5, x_1) = (q_{11}x_1, Y_1, D)$
 - glava ispituje postoji li poslije malog slova # ili novo malo slovo
- $\delta (q_5, \#) = ((p+1), \#, L)$
 - glava je pronašla # (postoji uzorak)

- $\delta((p+1), Y_1+1) = ((p+1), x_1+1, L)$
 - glava preskače sva velika slova i jedinice (da bi dodala jedinicu na krajnje lijevi dio trake)
- $\delta((p+1), B) = (q_0, 1, D)$
 - glava dodaje jedinicu na krajnje lijevi dio trake
- $\delta(q_{11}x_1, x_1) = (q_{11}x_1, x_1, D)$
 - glava preskače sva ostala slova desno od " x_1 " do # kada nađe slovo " x_1 "
- $\delta(q_{11}x_1, \#) = (q_{22}x_1, \#, D)$
 - glava je došla do # i prelazi u stanje $q_{22}x_1$
- $\delta(q_{22}x_1, (Y_n+0)) = (q_{22}x_1, (Y_n+0), D)$
 - glava preskače 0 i velika slova nakon #, dok ne nađe prvo malo slovo (prihvatljivo ili ne)
- $\delta(q_{22}x_1, x_1) = (q_3x_1, Y_1, L)$
 - glava je našla prihvatljivo malo slovo i označava ga (veliko slovo)
- $\delta(q_{22}x_1, (x_n - x_1)) = (q_{krivo}, (x_n - x_1), L)$
 - glava nije pronašla prihvatljivo malo slovo i prelazi u stanje q_{krivo}
- $\delta(q_{krivo}, Y_1) = (q_{krivo}, x_1, L)$
 - glava pretvara sva velika slova do prve 0 u mala slova
- $\delta(q_{krivo}, (0+\#)) = (q_{krivo1}, (0+\#), D)$
 - glava se vraća udesno na malo slovo i ide u q_{krivo1} kada dođe do 0 ili #
- $\delta(q_{krivo1}, x_1) = (q_{krivo2}, 0, L)$
 - glava pretvara bilo koje slovo (nakon 0 ili #) u 0
- $\delta(q_{krivo2}, 0) = (q_{krivo2}, 0, L)$
 - glava preskače sve 0 do #
- $\delta(q_{krivo2}, \#) = (q_{krivo3}, \#, L)$
 - glava nailazi na # i prelazi u q_{krivo3}

- $\delta (q_{krivo3}, x_1) = (q_{krivo3}, x_1, L)$
 - glava preskače sva mala slova
- $\delta (q_{krivo3}, Y_1) = (q_{krivo3}, x_1, L)$
 - glava pretvara velika slova u mala slova
- $\delta (q_{krivo3}, (1+B)) = (q_0, (1+B), D)$
 - glava onda ide u stanje q_0 kada nađe 1 ili B

2.2 Primjer

```

Upisi ulazni niz: ab#caab

d(q0,a)=>(q1a,A,D)
d(q1a,b)=>(q1a,b,D)
d(q1a,#)=>(q2a,#,D)
d(q2a,c)=>(q2a,θ,D)
d(q2a,a)=>(q3a,A,L)
d(q3a,θ)=>(q3a,θ,L)
d(q3a,#)=>(q4a,#,L)
d(q4a,b)=>(q4a,b,L)
d(q4a,A)=>(q5,A,D)
d(q5,b)=>(q11b,*,D)
d(q11b,#)=>(q22b,#,D)
d(q22b,θ)=>(q22b,θ,D)
d(q22b,A)=>(q22b,A,D)
d(q22b,a)=>(Qkrivo,a,L)
d(Qkrivo,A)=>(Qkrivo,a,L)
d(Qkrivo,θ)=>(Qkrivo1,θ,D)
d(Qkrivo1,a)=>(Qkrivo2,θ,L)
d(Qkrivo2,θ)=>(Qkrivo2,θ,L)
d(Qkrivo2,#)=>(Qkrivo3,#,L)
d(Qkrivo3,*)=>(Qkrivo3,b,L)
d(Qkrivo3,A)=>(Qkrivo3,a,L)
d(Qkrivo3,B)=>(q0,B,D)
d(q0,a)=>(q1a,A,D)
d(q1a,b)=>(q1a,b,D)
d(q1a,#)=>(q2a,#,D)
d(q2a,θ)=>(q2a,θ,D)
d(q2a,θ)=>(q2a,θ,D)
d(q2a,a)=>(q3a,A,L)
d(q3a,θ)=>(q3a,θ,L)
d(q3a,θ)=>(q3a,θ,L)
d(q3a,#)=>(q4a,#,L)
d(q4a,b)=>(q4a,b,L)
d(q4a,A)=>(q5,A,D)
d(q5,b)=>(q11b,*,D)
d(q11b,#)=>(q22b,#,D)
d(q22b,θ)=>(q22b,θ,D)
d(q22b,θ)=>(q22b,θ,D)
d(q22b,A)=>(q22b,A,D)
d(q22b,b)=>(q3b,*,L)
d(q3b,A)=>(q3b,A,L)
d(q3b,θ)=>(q3b,θ,L)
d(q3b,θ)=>(q3b,θ,L)
d(q3b,#)=>(q4b,#,L)
d(q4b,*)=>(q5,*,D)
d(q5,#)=>(p+1,#,L)
d(p+1,*)=>(p+1,b,L)
d(p+1,A)=>(p+1,a,L)
d(p+1,B)=>(q0,1,D)
d(q0,a)=>(q1a,A,D)
d(q1a,b)=>(q1a,b,D)
d(q1a,#)=>(q2a,#,D)
d(q2a,θ)=>(q2a,θ,D)
d(q2a,θ)=>(q2a,θ,D)
d(q2a,A)=>(q2a,θ,D)
d(q2a,*)=>(q2a,θ,D)
d(q2a,B)=>(qe,B,L)

vremenska složenost ucitanog niza je: 56
prostorna složenost ucitanog niza je: 9

1Ab#000[Qe]0B

Press any key to continue . . .

```

Kao primjer, navesti ću ulazni niz (slika 2):

ab#caab

U tom nizu, "ab" je uzorak, a "caab" tekst. Odmah se može vidjeti da će broj pojavljivanja uzorka u tekstu biti samo 1, s obzirom da je primjer trivijalan. Taj konačni rezultat (1) se upravo ova jedinica na dnu slike prije "press any key to continue...".

U privitku (*izlaz.txt* i programsko rješenje kako sam došao do ispisa *ispis_na_traku.c*) sam stavio i ispis trake za isti primjerni niz:

ab#caab

U tom ispisu se može točno vidjeti kako se glava Turingovog stroja kretala i kako je provjeravala slova u tekstu.

slika 2 – primjer ulaznog niza

3. Zaključak

Turingov stroj (TS) je najopćenitiji matematički model računanja koji se sastoji od upravljačke jedinice koja uz pomoć glave za čitanje i pisanje, čita i zapisuje znakove sa i na ulaznu traku.

Zadatak ovog projekta je bio da se pomoću Turingovog stroja prebroji pojava uzoraka u nekom tekstu. Programsko rješenje je napravljeno u C-u, a tablica prijelaza je poprilično opsežna. Ona ponajprije ovisi o ograničenjima, tj o broju znakova kojima se može uzorak napisati.

Princip rada je takav da glava Turingovog stroja radi na način da si označava slova koja je prošla i pregledala te ih tako "pamti" (iako ne doslovno jer nema nikakve memorije). Nakon što jedno slovo zapamti, ide provjeravati drugo i tako sve dok postoje slova uzorka.

4. Literatura

- http://hr.wikipedia.org/wiki/Turingov_stroj
- *Uvod u teoriju računarstva*, Siniša Srbljić, Element, Zagreb, 2007. godina