Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Larper

**Druga domaća zadaća iz predmeta  
„Uvod u teoriju računarstva“**

Zadatak broj 3050

Zagreb, lipanj 2011.

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

**Student:** Larper

**Matićni broj studenta:** 123456789

**Zadatak broj: 3050:**

Definirati (nije potrebno programski ostvariti) Turingov stroj koji zadane ulazne podatke sažima metodom LRE kompresije (Length Run Encoding). Ćelija ulazne trake smatra se jednim nibbleom čija se vrijednost može kretati u rasponu 0–15 (preporuča se uporaba heksadekadskog zapisa). Turingov stroj treba slijedni zapis podataka prevesti u oblik u kojem će se za svaki podatak zapisati njegova vrijednost i broj uzastopnih ponavljanja na traci.

Primjer:

Podaci prije kompresije:

1125555AAAAA555FFF

Podaci nakon kompresije:

122154A553F2

**1. Uvod**

**1.1 Turingov stroj**

Turingov stroj je apstraktni uređaj i predstavlja najopćenitiji matematički model računanja. Osnovni model Turingovog stroja sastoji se od upravljačke jedinke i ulazne trake. Upravljačka jedinka se nalazi u jednom od konačnog broja stanja i na temelju ulaznog znaka sa trake prelazi u novo stanje, zapisuje neki znak na traku i pomiče glavu za čitanje i pisanje u lijevo ili desno ili je ne miče.

Postoje prošireni modeli Turingovog stroja koji služe za jednostavniji rad pri rješavanju nekog problema. Jedan od takvih modela je Turingov stroj s višestrukim trakama. On se također sastoji od upravljačke jedinke, ali ima više glava za čitanje i pisanje, na svakoj traci po jednu. Upravljačka jedinka donosi odluku na temelju stanja upravljačke jedinke i n pročitanih znakova sa n traka. Jednim prijelazom Turingov stroj promijeni stanje, zapiše n znakova na n traka i bilo koju od n glava pomakne nezavisno u lijevo ili desno ili je ne pomakne. Traka na kojoj se nalazi ulazni niz naziva se: „ulazna traka“, a sve ostale: „radne trake“.



**Slika 1**. Model TS s višestrukim trakama

Turingov stroj se formalno defnira uređenom sedmorkom ts(Q,∑,Γ,δ,q0,B,F) gdje je:

Q - konačan skup stanja;

∑ - konačan skup ulaznih znakova;

Γ - konačan skup znakova trake;

δ - konačan skup funkcija prijelaza;

q0 - početno stanje;

B - znak kojim se označava prazna ćelija;

F - skup prihvatljivih stanja.

**1.2 Length Run Encoding**

Length run encoding ili češći naziv: Run-Length Encoding (RLE) je vrlo jednostavan oblik kompresije podataka u kojem se podnizovi koji se sastoje od jednog te istog znaka zamjenjuju s dva znaka: vrijednost i broj pojavljivanja. Ovaj način kompresije je najkorisniji nad podacima koji se sastoje od velikog broja takvih podnizova. Npr. jednostavni slikovni prikazi, crteži i animacije. Manje je koristan za podatke koji se ne sastoje od velikog broja takvih podnizova jer bi se veličina podatka mogla bitno povećati.

Primjer:

WWWABBBB222222222211111111

Ako na ovaj niz primjenimo RLE kompresiju dobit ćemo sljedeće:

W3A1B421018

Originalni niz sastoji se od 26 bajtova, a kompresiran niz od 11 bajtova. Što znači da je kompresiran podatak u veličini bajtova manji za 2,36 puta.

**2. Ostvarenje**

**2.1 Ideja rješenja**

U zadatku je potrebno ulazni niz znakova kompresirati metodom RLE kompresije. Turingov stroj koji riješava ovaj problem imati će ulaznu traku i jednu radnu traku. Na radnoj traci biti će zapisan kompresiran niz kada Turingov stroj završi s radom. Zbog jednostavnosti zapisa koristiti se heksadekaski zapis kao što je preporučeno u zadatku. Pošto se jedna ćelija na traci definira kao jedan nibble (4 bita), najveći mogući broj koji se može zapisat je 00002 (Pošto je u zadatku u primjeru zadano da se jedno pojavljivanje označava znakom 1, to znači da se znak 0 ne koristi u brojaču. Zato da se poboljša kompresija, 0 predstavlja broj 1610). Tu se javlja problem brojača znakova tj. kako ćemo razlikovati vrijednost znaka od njegovog broja pojavljivanja i također, javlja se problem zapisivanja brojača za broj pojavljivanja veći od 0. Taj problem će se riješiti cirkularnim brojačem. Kada broj pojavljivanja premaši 0, zapiše se znak i trenutni broj pojavljivanja tj. 0 i nastavi se brojati.

Primjer:

AAAAAAAAAAAAAAAAA

U ovom nizu imamo znak A koji se pojavljuje 17 puta.

Kompresiran niz izgleda ovako:

A0A1

**2.2 Definicija i rad Turingovog stroja**

Turingov stroj koji riješava ovaj problem ima 34 stanja: jedno početno stanje, jedno prihvatljivo stanje, 16 stanja - po jedno za svaki znak i još 16 stanja – po jedno za svaki znak, koja služe da znamo kad zapisujemo broj ponavljanja znaka na radnu traku. TS je zadan uređenom sedmorkom TS(Q, ∑, Γ, δ, qP, #, {qZ}) gdje je:

Q = {qP, qZ, q0, ..., qF, q0W, ..., qFW};

∑ = {0, 1, ..., F};

Γ = {0, 1, ..., F, #};

qP je početno stanje;

# je oznaka prazne ćelije;

qZ je završno i ujedno prihvatljivo stanje TS.

Funkcije prijelaza izgledaju na sljedeći način:

δ(q, [X1, X2]) -> (p, [Y1, Y2], [Z1, Z2])

TS iz nekog stanja q, na ulaznoj traci pročita znak X1, a na radnoj traci pročita znak X2. TS prelazi u stanje p, na ulaznu traku zapisuje znak Y1, a na radnu traku zapisuje znak Y2. Z1 i Z2 označavaju u koju stranu se pomiču respektivne glave. Z1 i Z2 su elementi iz skupa {R, S} gdje R znači pomak u desno, a S znači da glave ostaje na mjestu.

U nastavku će biti opisan rad i funkcije prijelaza TS.

Glava na ulaznoj traci nalazi se na ćeliji u kojoj je prvi znak niza. Glava na radnoj traci nalazi se na nekoj ćeliji koja je prazna. Turingov stroj započinje s radom sa sljedećim skupom funkcija prijelaza:

δ(qP, [X, #]) -> (qX, [X, X], [S, R])

X = {0, 1, ..., F}

TS se nalazi u početnom stanju. Glava na ulaznoj traci pročita neki znak X, a glava na radnoj traci pročita praznu ćeliju. TS prelazi u stanje koje predstavlja taj znak – qX, na ulaznu i radnu traku zapisuje taj isti znak, glava na ulaznoj traci ostaje na mjestu, glava na radnoj traci pomiče se u desno. TS nastavlja rad sa sljedećim skupom funkcija prijelaza:

δ(qX, [X, #]) -> (qXW, [X, 1], [R, S])

X = {0, 1, ..., F}

TS se nalazi u nekom stanju qX koje predstavlja koji je znak u prethodnom koraku bio pročitan. Glava na ulaznoj traci pročita taj isti znak X, a glava na radnoj traci pročita praznu ćeliju. TS prelazi u stanje qXW koje označava da će na radnu traku početi pisati broj pojavljivanja znaka X. Na ulaznu traku TS zapisuje znak X, a na radnu traku zapisuje 1 jer se znak X za sad pojavio samo jednom. Glava na ulaznoj traci se pomiče u desno, a glava na radnoj traci se ne miče; jer ako je sljedeći znak opet X TS će morati na isto mjesto zapisat sljedeći broj pojavljivanja. TS nastavlja rad sa sljedećim skupom funkcija prijelaza:

δ(qXW, [X, Y - 1]) -> (qXW, [X, Y], [R, S])

X = {0, 1, ..., F}

Y = {2, 3, ..., F, 0}

TS se nalazi u nekom stanju qXW koje predstavlja za koji se znak u prethodnom koraku brojač znakova uvećao za jedan. TS s ulazne trake pročita taj isti znak X, a s radne trake pročita broj pojavljivanja u prethodnom koraku. TS prelazi u stanje qXW, na ulaznu traku zapisuje X, a na radnu traku zapisuje broj pojavljivanja uvećan za jedan. Glava na ulaznoj traci se pomiče u desno, a glava na radnoj traci ostaje na mjestu.

Sljedeći skup funkcija prijelaza opisuje rad TS kada brojač znakova premaši 0 tj. 1610:

δ(qXW, [X, 0]) -> (qP, [X, 0], [S, R])

X = {0, 1, ..., F}

TS se nalazi u nekom stanju qXW koje predstavlja za koji se znak u prethodnom koraku brojač znakova uvećao za jedan. TS s ulazne trake pročita taj isti znak X, a sa radne trake pročita 0 tj. 1610. To znači da je za tu ćeliju brojač došao da maksimuma. TS prelazi u početno stanje, na ulaznu traku zapisuje znak X, a na radnu traku zapisuje znak 0. Glavu na ulaznoj traci ne miče, a glavu na radnoj traci pomiče u desno u kojoj je sada prazna ćelija. Time je spreman opet krenuti brojati taj znak X postupkom objašnjenim s prijašnjim funkcijama prijelaza.

Sljedeći skup funkcija prijelaza opisuje rad TS kada iz nekog stanja qXW pročita na ulaznoj traci znak Y koji je različit od X:

δ(qXW, [Y, Z]) -> (qP, [Y, Z], [S, R])

X = {0, 1, ..., F}

Y = {0, 1, ..., F}

Z = {1, 2, ..., F, 0}

TS se nalazi u stanju qXW koje predstavlja za koji se znak u prethodnom koraku brojač znakova uvećao za jedan. TS s ulazne trake pročita znak Y koji je različit od znaka X, a s radne trake pročita brojač Z koji predstavlja koliko se puta pojavio znak X do sada. TS prelazi u početno stanje, na ulaznu traku zapisuje znak Y, a na radnu traku zapisuje nazad brojač Z. Glava na ulaznoj traci ostaje na mjestu, a glava na radnoj traci se miče u desno na praznu ćeliju. S ovim skupom funkcija prijelaza TS je u sljedećem koraku spreman brojati broj pojavljivanja znaka Y postupkom koji je već opisan.

Sljedeći skup funkcija prijelaza opisuje rad TS kada iz nekog stanja qXW pročita na ulaznoj traci znak # tj. kada TS dođe do kraja niza:

δ(qXW, [#, X]) -> (qZ, [#, X], [S, R])

X = {1, 2, ..., F, 0}

TS se nalazu u nekom stanju qXW koje predstavlja za koji se znak u prethodnom koraku brojač znakova uvećao za jedan. TS s ulazne trake pročita znak #, a s radne trake pročita brojač znakova X. Pošto je TS došao do kraja niza, TS prelazi u završno (prihvatljivo) stanje, na ulaznu traku zapisuje #, a na radnu traku nazad zapisuje brojač znakova X. Glava na ulaznoj traci se ne miče, a glava na radnoj traci se pomiče u desno na praznu ćeliju. Pošto stanje qZ nema definiranu funkciju prijelaza niz se prihvaća tj. TS je završio s kompresiranjem niza.

**3. Zaključak**

**3.1 Zaključak**

Turingov stroj iako je jednostavan, omogućava nam da njime obavljamo komplicirane funkcije. Međutim, relativno jednostavnu operaciju, kao što je slučaj u ovoj zadaći, nije moguće ostvariti malim brojem stanja, znakova i funkcija prijelaza. U ovom konkretnom slučaju imamo preko 500 funkcija prijelaza jer TS za svaki slučaj mora imati posebnu funkciju prijelaza.

RLE kompresija iako je jednostavna, omogućava nam da izrazito efikasno kompresiramo podatke. Pogotovo ako ti podaci imaju jako puno podnizova uzastopnih ponavljanja jednog te istog znaka. U ovom zadatku problem se javio pri implementaciji RLE kompresije Turingovim strojem, pošto je zadano da je jedna ćelija na traci jedan nibble, taj problem je riješen na prilično jednostavan način; sa cirkularnim brojačem znakova.

Zadatak je riješen tako da TS počne čitati niz znakova i čim pročita prvi znak on prelazi u stanje koje definira taj znak. Nakon toga TS zapisuje taj znak na radnu traku i prelazi u stanje koje označava da će početi pisati broj pojavljivanja tog znaka na radnu traku. Na taj način on će zapisivati broj pojavljivanja tog znaka sve dok brojač ne dođe do 0 ili TS pročita znak s ulazne trake koji je različit od znaka za kojeg se povećava brojač ili TS s ulazne trake pročita praznu ćeliju. U prva dva slučaja to je riješeno tako da TS prelazi u početno stanje i počinje brojati iznova, a u trećem slučaju TS prelazi u završno (prihvatljivo) stanje i prestaje s radom.

**4. Literatura**

**4.1 Popis literature**

1. Uvod u teoriju računarstva, Srbljić S., Element, Zagreb, 2007.

2. Run-Length Encoding (RLE). <http://www.fileformat.info/mirror/egff/ch09_03.htm>(5.6.2011.)

3. Run-length encoding <https://secure.wikimedia.org/wikipedia/en/wiki/Run_length_encoding>(5.6.2011.)