Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Stjepan Antivo Ivica

1. seminarski rad - Samostalni studentski projekt iz predmeta "Uvod u teoriju računarstva"

Zadatak broj 3028

Zagreb, lipanj 2011.

Stjepan Antivo Ivica

0036448356

Zadatak broj 3028: U HTML –u je upisana tablica sa nekim podacima. Potrebno je razviti model automata (DKA, NKA, potisnog automata ili Turingovog stroja – prema potrebi za prepoznavanje jezika) koji bi provjeravao ispravnost upisane tablice. Model implementirati u nekom programskom jeziku (eksplicitnim načinom zapisa automata). Moguća je pojava samo slijedećih oznaka <TABLE>, </TABLE>, <TD>, </TD>, <TR>, </TR>. Pretpostaviti da oznake za tablice nemaju dodatnih parametara i da u svakom retku ima jednak broj stupaca.

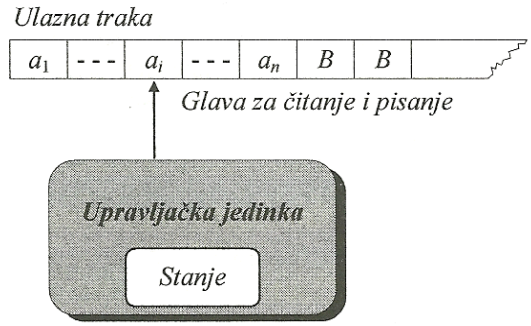
Napomena:

* automat ne mora paziti je li u svakom retku jednak broj stupaca i ne mora paziti da oznake elemenata nemaju atribute

1. Uvod

1.1 Turingov stroj

Turingov stroj predstavlja najopćenitiji matematički model računanja te je istih mogućnosti kao i bilo koje digitalno računalo. Osnovna primjena Turingovog stroja je u prihvaćanju jezika. Postoji mnogo različitih modela Turingovog stroja koji su svi izvedeni iz osnovnog modela Turingovog stroja. Osnovni model Turingovog stroja sastoji se od upravljačke jedinke, glave za čitanje i pisanje, kao i ulazne trake. Upravljačka jedinka se može nalaziti u jednom od konačnog broja stanja, koja se dijele na prihvatljiva i neprihvatljiva stanja. Za razliku od konačnih automata i potisnih automata Turingov stroj ima mogućnost pisanja po ulaznoj traci. Traka sa koje Turingov stroj može čitati i pisati je ograničena s lijeve strane, dok je s desne strane beskonačna. Kod Turingovog stroja glava za čitanje i pisanje ima mogućnost pomaka u desnu ili lijevu stranu, osim u slučaju kada se glava Turingovog stroja nalazi na krajnje lijevoj ćeliji, tada ju je moguće pomaknuti samo u desno.



Slika 1. *Osnovni model Turingovog stroja*

Turingov stroj zadaje se 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 - funkcija prijelaza;

q0 - početno stanje;

B - znak kojim se označava prazna ćelija;

F - skup prihvatljivih stanja

Dozvoljeno je da funkcija prijelaza δ 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 ($W \in \{L,R\}$). Ukoliko se Turingov stroj zaustavio u stanju koje pripada skupu prihvatljivih stanja, ulazni niz se prihvaća, u suprotnom se ulazni niz ne prihvaća.

Jedan od proširenih modela Turingovog stroja je Turingov stroj s višestrukim trakama. Ovaj model Turingovog stroja ima k glava za čitanje i pisanje i k traka (koje mogu biti beskonačne s obje strane, ili ograničene s jedne strane). Upravljačka jedinka Turingovog stroja donosi odluku na temelju dviju grupa parametara, stanja upravljačke jedinke i k pročitanih znakova sa k traka. U jednom prijelazu Turingov stroj promijeni stanje, zapiše k znakova na k traka i pomakne bilo koju od k glava nezavisno u desno, lijevo ili ju ne pomakne. Na jednu traku, koja se naziva ulazna traka, zapisan je niz koji se ispituje. Sve ostale trake se nazivaju radnim trakama.

1.2. HTML tablica

HTML (kratica za HyperText Markup Language) je programski jezik koji se koristi za kreiranje dokumenata na World Wide Web-u. HTML se koristi za stvaranje hipertekstualnih datoteka (datoteka koje sadržavaju linkove) .

HTML tablica definira se izmeu para HTML tagova <TABLE> i </TABLE>. Tablica može sadržati proizvoljan broj redaka omeđenih <TR> i </TR> tagovima. U svakom retku nalazi se jednak broj ćelija omeđenih sa <TD> i </TD> (podatkovne ćelije).

2. Ostvarenje

2. 1. Koncept ideje i model Turingovog stroja

Za rješenje zadanog zadatka odabran je Turingov stroj sa jednom ulaznom trakom i jednom radnom trakom. Pri tome se HTML tablica zapisana na ulaznu traku, a radna traka se koristi za daljnje provjere. Obje trake su desno beskonačne. Na radnoj traci se prije početka samog rada stroja nalazi sljedeći znak: <, a traka je pozicionirana na prvo desno polje praznine (drugo polje radne trake). Na ulaznoj traci se prepoznaju specifične oznake (<TABLE>, </TABLE>, <TR>, </TR>, <TD>,</TD>) kao i ostali nizovi znakova. Tako prepoznati dijelovi tablice se zapisuju na radnu traku u obliku kraće oznake na sljedeći način:

T – oznaka za <TABLE>

/> - oznaka za </TABLE>

R – oznaka za <TR>

X – oznaka za </TR>

D – oznaka za <TD>

Y – (oznaka za </TD>)

W – oznaka za niz znakova koji nije specifična oznaka

Provjere ispravnosti koje se vrše na ulaznoj traci su odbacivanje ulaznog niza koji na svom početku ne počinje sa znakom < te ako ne završava sa </TABLE>. Ukoliko niz nije odbačen, a svi znakovi ulazne trake su pročitani sav daljnji rad će se vršiti na radnoj traci pri kojem će se slijedeći određena pravila odlučiti o prihvatljivosti zadane tablice.

Ovaj Turingov stroj je zadan formalnom definicijom:

TS H=(Q,Σ,Γ,δ, q0, B,F)

Q = {qSTART, qS, qW, qZ, qT, qA, qB, qL, qE, q/, q/T, q/A, q/B, q/L, q/E, q>, qD, q/D, qR, q/R, qLIJEVO, qLL, qP, TABLE, ETABLE, qPRIH, q0, EROW, ECOLL, ROW, NIZ, COLL}

Σ = {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, <, >, /}

Γ = { A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, <, >, / , .}

δ – funkcija prijelaza ima ukupno petsto pedeset i pet, a njihov zapis se može naći u mapi gdje se nalazi ovaj rad u datoteci def.txt počevši od njenog šestog reda

q0 = qSTART

B = .

F = qPRIH

2. 2. Pojašnjenje rada TS H

Za lakše razumijevanje algoritma rada TS H taj algoritam će biti podijeljen u dva zasebna procesa.

U prvom procesu cilj je prepoznati skupine znakova i zapisati ih u sažetijem obliku na radnu traku.

Početno stanje u kojem se automat nalazi je qSTART. Iz kojeg postoji samo jedan prijelaz

qSTART, <, . -> qZ, ., ., R, S .

qS je stanje koje simbolizira da automat očekuje da na ulaznoj traci slijedi ili proizvoljan niz znakova ili specifična oznaka. Ukoliko se ostvari prijelaz

qS, ., . -> qL, ., ., S, L

to znači da je pročitana cijela ulazna traka te da treba započeti rad na pomoćnoj traci. Prijelazom

qS, <, . -> qZ, ., ., R, S

automat se upućuje da je možda naišao specifičnu oznaku I prebacuje u stanje za daljnju provjeru. Naiđe li u ovom stanju na radnoj traci na prazan znak, a na ulaznoj traci bilo koji drugi znak osim praznog znaka i znaka <, obavit ce se prijelaz poput sljedećeg

qS, A, . ->qW, ., W, R, R

qW je stanje koje označava da je automat upravo pročitao niz znakova te očekuje da još uvijek postoji niz znakova npr.

qW, A, . -> qW, ., ., R, S

ili da sljedeći niz znakova koje ce pročitati može označavati specifičnu oznaku

qW, <, . -> qZ, ., ., R, S

qZ je stanje u koje se podrazumijeva da je automat pročitao znak < te da postoji mogućnost da će biti pročitana specifična oznaku. Prijelazom

qZ, T, .-> qT, ., ., R, S

doznačeno nam je da je automat na ulaznoj traci pročitao niz znakova <T. Prijelazom

qZ, /, . -> q/ ,. ,., R, S

doznačeno nam je da je automat na ulaznoj traci pročitao niz znakova </. Dok prijelazima poput

qZ, A, . -> qW, ., W, R, R

automat podrazumijeva da niz znakova koji čita nije specifična oznaka.

Sada je jasna uloga stanja qA, qB, qL, qE koji odgovaraju pročitanom nizu znakova redom <TA, <TAB, <TABL, <TABLE. Sto nas upućuje I u svrhu stanja qR te qD koji odgovaraju pročitanom nizu znakova redom <TR te <TD.

Valja dodatno pojasniti prijelaze stanja qE čiji su prijelazi srodni prijelazima stanja qR te qD. Naravno prijelazi poput

qE, A, . -> qW, ., W, R, R

postoje kako ne bismo učitani niz znakova shvatili kao specifični tag. Dok prijelaz

qE, >, . -> qS, ., T, R, R

nam sa sigurnošću može tvrditi da je niz znakova koji smo upravo pročitali specifičan tag.

Sljedeći dosadašnja pojašnjenja zaključimo da stanje q/ odgovara pročitanom nizu znakova </ .

Upravo sada trebala biti jasna i uloga stanja q/A, q/B, q/L, q/E koji odgovaraju pročitanom nizu znakova redom </TA, </TAB, </TABL, </TABLE. Sto nas upućuje i u svrhu stanja q/R te q/D koji odgovaraju pročitanom nizu znakova redom </TR te </TD.

Prijelazi iz stanja q/R te q/D su srodni već spomenutim prijelazima iz stanja qE.

Dok prijelazi iz stanja q/E zahtijevaju dodatno pojašnjenje. Prijelaz

q/E, >, . -> q>, >, /, S, R

nam sa sigurnošću može tvrditi da je niz znakova koji smo upravo pročitali specifičan tag i to upravo </TABLE>. Naravno prijelazi poput

q/E, A, . -> qW, ., W, R, R

postoje kako ne bismo učitani niz znakova shvatili kao specifični tag.

Iz stanja q> postoji samo jedan prijelaz

q>, >, . -> qS, ., >, R, R

Tom prijelazu pojednostavljeno govoreći svrha je zapisati sljedeći znak > na radnu traku. Taj znak simbolizira da je tablica možda bila ugnježdena i pomoći će nam pri ispitivanju tog slučaja.

Kada su svi znakovi pročitani I ako se obavio već spomenuti prijelaz

qS, ., . -> qLIJEVO, ., ., S, L

tada se može nastaviti s drugim procesom u kojemu se sav rad odvija na radnoj traci.

Iz stanja qLIJEVO postoji samo jedan prijelaz

qLIJEVO, ., > -> qLL ,. ,., S, L

kojemu je zadatak skinuti posljednji znak > sa pomoćne trake. Razlog tomu je što posljednja specifična oznaka </TABLE> ne može zatvarati ugnježdenu tablicu već našu polaznu tablicu.

Iz stanja qLL postoji 9 prijelaza

qLL, ., > -> qLL,. , >, S, L

qLL, ., T -> qLL, ., T, S, L

qLL, ., / -> qLL, ., /, S, L

qLL, ., R -> qLL, ., R, S, L

qLL, ., X -> qLL, ., X, S, L

qLL, ., D -> qLL, ., D, S, L

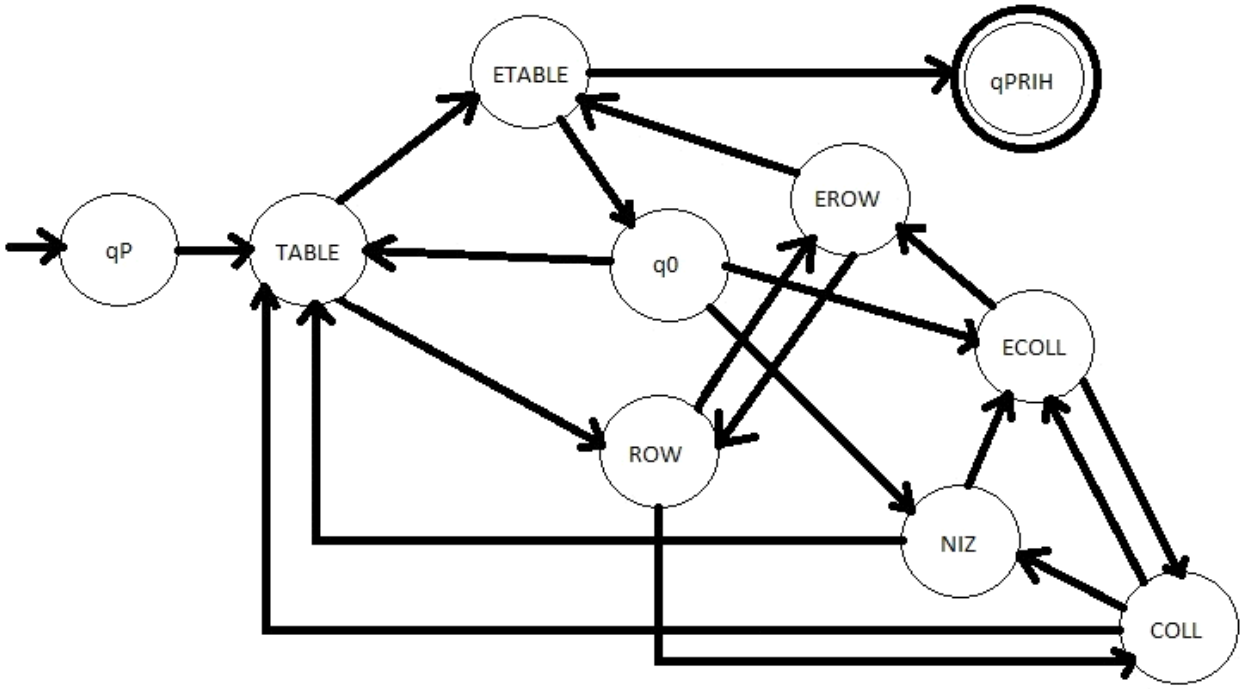
qLL, ., Y -> qLL, ., Y, S, L

qLL, ., W -> qLL, ., W, S, L

qLL, ., < -> qP, ., ., S, R

Razlog njegovog postojanja sada je jasan. Pojednostavljeno govoreći treba nam kako bismo pozicionirali glavu za čitanje i pisanje na početak radne te zatim na prvi znak nakon njenog početnog znaka.

Sljedeća slika pojašnjava koja su nam stanja i njihovi odnos bitni za nastavak rada na radnoj traci. Može se, a i prirodno jest zaključiti da je velik dio dosadašnjeg rada algoritma samo prethodio da bi se došlo do ovog pojednostavljenog zapisa na radnoj traci o čijoj ćemo prihvatljivosti biti u stanju sa izrazitom lakoćom odlučiti



Slika 2. *Dijagram koji pokazuje prijelaze između stanja po koji se smiju odvijati za niz na radnoj traci kako bi se prihvatila ulazna tablica*

Stanje qP je početno stanje koje se brine da tablica započinje specifičnom oznakom <TABLE> tj. njenom pojednostavljenom oznakom na radnoj traci T.

qP, ., T -> TABLE, ., ., S, R

Stanje qTABLE označava da je automat pročitao znak T (prevedeno oznaku <TABLE>). Prijelazi iz ovog stanja su

TABLE, ., R -> ROW, ., ., S, R

TABLE, ., / -> ETABLE, ., ., S, R

Sada su jasne i uloge stanja ETABLE, qPRIH, EROW, ECOLL, ROW, COLL. Ispišimo njihove prijelaze:

ROW, ., X -> EROW, ., ., S, R

ROW, ., D -> COLL, ., ., S, R

COLL, ., W -> NIZ, ., ., S, R

COLL, ., Y -> ECOLL, ., ., S, R

COLL, ., T -> TABLE, ., ., S, R

ECOLL, ., X -> EROW, ., ., S, R

ECOLL, ., D -> COLL, ., ., S, R

EROW, ., R -> ROW, ., ., S, R

EROW, ., / -> ETABLE, ., ., S, R

ETABLE, ., > -> q0, ., ., S, R

ETABLE, ., . -> qPRIH, ., ., S, R

Ostaje još pojasniti stanja q0 i NIZ. Stanje NIZ označava da je automat pročitao znak W koji označava niz znakova. Njegovi prijelazi su:

NIZ, ., W -> NIZ, ., ., S, R

NIZ, ., T -> TABLE, ., ., S, R

NIZ, ., Y -> ECOLL, ., ., S, R

Stanje q0 označava da je automat pročitao znak > koji označava da je tablica bila ugnježđena.

q0, ., T -> TABLE, ., ., S, R

q0, ., W -> NIZ, ., ., S, R

q0, ., Y -> ECOLL, ., ., S, R

3. Zaključak

Pri rješavanju zadanog problema korišten je Turingov stroj s više traka, preciznije s dvije trake, od kojih je jedna ulazna traka, a druga je radna traka. Definirani Turingov stroj provjerava ispravnost HTML tablice. Ideja po kojoj funkcionira jest da niz na ulaznoj traci koji predstavlja HTML tablicu pretvori u pojednostavljeni zapis na radnoj traci gdje vrši daljnja ispitivanja.

Ovakvim postupkom postignut je jako brz i efikasan algoritam. Naime za nizove koji se ne odbacuju odmah potrebno je obaviti maksimalno 3n prijelaza , gdje je n broj znakova na ulaznoj traci (očekivanje autora ovog teksta je da je ta ovisnost manja od 1.5n, no jasno je da se radi o linearnoj ovisnosti). Razlog tomu je što znakove na ulaznoj traci treba samo jednom pročitati i zapisivati ulazni niz na radnoj traci u obrađenom (kraćem) obliku te na radnoj traci jednom vratiti se u lijevo te nastaviti čitati u desno.

Algoritam se može poboljšati smanjenjem broja stanja, ali tada njegov opis ne bi bio jednako transparentan.

Algoritam može procijeniti ispravnost zapisa HTML tablice koja može sadržavati sva velika slova engleske abecede.

Algoritam se može proširiti za mala slova engleske abecede kao i za ostale znakove koji nisu velika slova engleske abecede.

Postoje i drugi algoritmi koji se mogu primijeniti na zadani problem. Ovaj je odabran među nekolicinom ideja zbog njegove brzine i jednostavnosti.

Kao što vidimo Turingov stroj je vrlo moćan matematički model koji je u stanju obavljati složene operacije koje mogu obavljati računala.

4. Literatura

4. 1. Popis literature

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