Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Andrija Čajić

# Druga domaća zadaća iz predmeta

# “Uvod u teoriju računarstva”

Zadatak broj 2028

Zagreb, lipanj 2009.

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

**Student:** Andrija Čajić

**Matični broj studenta:** 0036440342

**Zadatak broj 2028:** Napisati gramatiku i izgraditi potisni automat koji će prihvaćati ispravno napisane aritmetičke izraze. Izrazi uključuju operatore zbrajanja, oduzimanja, množenja i dijeljenja, znakove jednakosti, nejednakosti, i usporedbe (manje, veće, manje ili jednako i veće ili jednako) te zagrade. Primjer ispravno napisanih izraza:

(a<b)

((a-b+f)!=c\*e)

NAPOMENA: U izrazima se mogu pojavljivati samo varijable. Varijable moraju započinjati slovom.

## Uvod

Potisni automat gradi se za potrebe prihvaćanja kontekstno neovisnih jezika. Postoje kontekstno neovisni jeziki koji nisu regularni i zato je potrebno proširiti model konačnog automata da bi se dobio potisni automat. Upravljačkoj jedinki konačnog automata dodaje se potisni stog (LIFO stog). Potisni automat ima glavu za čitanje i ulaznu traku jednako kao i konačni automati, no ima i potisni stog. Osim čitanja znakova ulazne trake, upravljačka jedinka čita i jedan znak sa vrha potisnog stoga. Nakon čitanja znaka, s vrha stoga uzima se pročitani znak, a na vrh stoga stavlja se niz znakova. Upravljačka jedinka je u jednom od konačnog broja stanja. Upravljačka jedinka donosi odluku o promjeni sadržaja vrha stoga, pomaku glave za čitanje i promjeni stanja. Odluka se donosi na temelju tri podatka:

1. stanja upravljačke jedinke
2. znaka koji je na vrhu stoga
3. znaka na ulaznoj traci.

Moguće je da upravljačka jedinka donese odluku na temelju samo dva podatka: stanja upravljačke jedinke i znaka na vrhu stoga. Npr. δ(q, ε, K) = {(p, IK)}. U ovom primjeru automat će dok je u stanju q i kad je na vrhu stoga znak K, moći bez učitanog znaka prijeći u stanje p i na vrh stoga dodati još jedan znak – I. Budući da se odluka donsi bez čitanja znaka ulazne trake, glava za čitanje ne miče se za jedno mjesto u desno. Upravljačka jedinka koji se niz stavlja na vrh stoga. Moguće je staviti prazan niz ε (ekvivalentno skidanju znaka sa vrha stoga), jedan znak (zamijeni vrh stoga sa tim novim znakom), niz od više znakova (zamijeni vrh stoga nizom znakova).

Potrebno je napisati gramatiku G = (V, T, P, S) koja će generirati ispravne aritmetičke izraze i izgraditi potisni automat PA M = (Q, Σ, Γ, δ, q0, Z0, F) koji će prihvaćati sve ispravno napisane aritmetičke izraze.

## Ostvarenje

Početni nezavršni znak gramatike je S. Kao što je već spomenuto, gramatika treba generirati aritmetički izraz. Taj aritmetički izraz može biti unutar zagrada ili ne. Zato postavljam prve dvije produkcije:

S → I | (I)

Nezavršni znak I predstavlja izraz. Izraz može biti oblika: vrijednost\_usporedba\_ vrijednost ili samo vrijednost. Zato stvaram produkcije:

I → P=P | P!=P | P>P | P<P | P>=P | P<=P | P

Nezavršni znak P predstavlja vrijednost. Vrijednost je usko povezana sa elementom. Stvaram produkcije:  
P → -C | C | +C

Nezavršni znak C predstavlja element. Razdvajanje pojmova vrijednosti i elementa služi isključivo za reguliranje problema predznaka kod nekih varijabli u aritmetičkim izrazima. Stvaram produkcije za element:

C → C+C | C-C | C\*C | C/C | (P) | X

Četiri produkcije elementa zadane su rekurzivno. To omogućuje širenje aritmetičkog izraza na više varijabli. Element ne dopušta postavljanje predznaka samom sebi (npr. C -> -C) jer bi to značilo potencijalno nagomilavanje operatora što nije poželjno. Za takve stvari potrebno je otvoriti zagradu što čini produkcija C → (P). Ta produkcija između ostalog omogućuje nizanje predznaka ali na ispravan način, recimo, -(+(-(a))). Konačno, element može producirati X, odnosno neku konkretnu varijablu.

X → lY | l

Završni znak l (letter) predstavlja znak abecede. To je prvi znak imena varijable koje može imati više znakova. Prvi znak okarakteriziran je time što je ograničen da bude slovo, a ne primjerice broj ili neki drugi znak. Nastavak znakova imena varijable je Y.  
Y → lY | dY | l | d

Završni znak d (digit) predstavlja znamenku. Nastavak imena varijable, prema ovim produkcijama, može sadržavati proizvoljan broj znakova (slova i znamenki).

Ako sumiramo sve dosad zapisane produkcije, dobijemo gramatiku G' = ({I, P, C, X, Y, S}, {(, ), =, !, >, <, +, -, \*, /, l, d}, P, S). Produkcije P su:

S → I | (I)

I → P=P | P!=P | P>P | P<P | P>=P | P<=P | P

P → -C | C | +C

C → C+C | C-C | C\*C | C/C | (P) | X

X → lY | l

Y → lY | dY | l | d

Ovu gramatiku izmjenit ću u namjeri da pojednostavim izgradnju potisnog automata koji će prihvaćati jezik koji generira ova gramatika G = ({I, P, C, X, Y, S, Z, J, U, V, O}, {(, ), =, !, >, <, +, -, \*, /, l, d}, P, S). Produkcije P su:

S → I | (IZ

Z → )

I → PJP | PUJP | PVP | PVJP | P

J → =

U → !

V → > | <

P → -C | C | +C

C → COC | (PZ | X

O → + | - | \* | /

X → lY | l

Y → lY | dY | l | d

Izmjene koje su napravljene u biti se svode na sljedeće:

* desne strane produkcija imaju jedan završni znak (u potisnom automatu za učitani znak skida znak sa stoga)
* desne strane produkcija imaju jedan završni znak i više nezavršnih znakova i završni znak je na krajnje lijevom mjestu (u potisnom automatu za učitani znak zamjenjuje vrh stoga sa nizom znakova)
* desne strane produkcija nemaju niti jedan završni znak (u potisnom automatu ε-prijelazom vrh stoga zamjenjuje nizom znakova)

Izgradnja potisnog automata koji prihvaća praznim stogom zapravo je analogna izmjenjenoj verziji gramatike.

PA M = ({q}, {+, -, \*, /, !, =, >, <, (, ), l , d}, {I, Z, P, J, U, V, C, O, X, Y, K}, δ, q, K, Ø)

Stvaram funkcije prijelaza na temelju produkcija gramatike.

δ(q, ε, K) = {(q, IK), (q, ε)}

δ(q, (, K) = {(q, IZK)}

δ(q, ε, I) = {(q, PJP), (q0, PUJP), (q0, PVP), (q0, PVJP), (q0, P)}

δ(q, -, P) = {(q, C)}

δ(q, +, P) = {(q, C)}

δ(q, ε, P) = {(q, C)}

δ(q, ε, C) = {(q, COC), (q, X)}

δ(q, (, C) = {(q, PZ)}

δ(q, l, X) = {(q, Y), (q, ε)}

δ(q, l, Y) = {(q, Y), (q, ε)}

δ(q, d, Y) = {(q, Y), (q, ε)}

δ(q, ), Z) = {(q, ε)}

δ(q, +, O) = {(q, ε)}

δ(q, -, O) = {(q, ε)}

δ(q, \*, O) = {(q, ε)}

δ(q, /, O) = {(q, ε)}

δ(q, =, J) = {(q, ε)}

δ(q, !, U) = {(q, ε)}

δ(q, >, V) = {(q, ε)}

δ(q, <, V) = {(q, ε)}

## Zaključak

U ovom seminarskom radu opisan je način stvaranja gramtike i izgradnje potisnog automata koji prihvaća ispravno napisane aritmetičke izraze. Prilikom pokušaja formiranja gramatike bilo je nužno izbjeći moguće propuste u smislu da gramatika nije smjela generirati jezik koji nije aritmetički izraz, a ujedno je morala biti u mogućnosti generirati sve ispravno napisane aritmetičke izraze. Da bi se to postiglo, pravila zapisa aritmetičkog izraza morala su biti jasno i jednoznačno definirana. Radi jednostavnosti automata ta pravila su ujedno morala biti definirano koncizno, u što manje načela. Ovako izgrađena gramatika i potisni automat vjerojatno nisu optimirani i ostvarivi su možda u manjem broju produkcija odnosno manjem broju funkcija prijelaza. Ovaj potisni automat je nederministički potisni automat zato što ima definirane prijelaze iz stanja bez čitanja znaka sa ulazne trake i ima definirane prijelaze sa čitanjem znaka sa ulazne trake za isto stanje i isti vrh stoga. Npr. δ(q, +, P) = {(q, C)} i δ(q, ε, P) = {(q, C)}. Također postoje postoje skupovi δ(q, a, Z) koji imaju više elemenata, gdje je a bilo koji ulazni znak iz Σ ⋃ {ε}, a Z bilo koji znak stoga iz Γ. Npr. δ(q, ε, I) = {(q, PJP), (q0, PUJP), (q0, PVP), (q0, PVJP), (q0, P)}. Potisni automat ima samo jedno stanje zato jer mu više niti ne treba. Bitna stvar koji automat koristi je stog koji mu u svakom trenutku izvođenja govori zapravo, laički rečeno, što bi još automat u daljnjem radu trebao učitati sa ulazne trake da bi niz bio prihvatljiv. Logično, ako je stog prazan i učitana je cijela ulazna traka, automatu ne treba više ništa da bi prihvatio niz pa ga zato i prihvati. U suprotnom, ako automat zaustavi rad (nije pronašao funkciju prijelaza za trenutno stanje, ulazni znak i znak na vrhu stoga ili je pročitao zadnji znak na ulaznoj traci), a stog nije prazan, niz se ne prihvaća jer automat više ne može učitati ono što mu treba kako bi prihvatio niz.