
REŠENJE

Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programske Prevodioci 1 (IR4PP1)

Nastavnik: doc dr Dragan Bojić

Asistent: dipl ing Miloš Gligorić

Ispitni rok: Jun 2008.

Datum: 12.07.2008.

Napomene:

- *Ispit traje 3 sata. Prvih 30 minuta nije dozvoljeno napuštanje sale. Prva dva sata nije dozvoljeno iznositi ispitne zadatke. Poslednjih 10 minuta ispita nije dozvoljeno napuštanje sale.*
- *Na ispitnoj vežbanci obavezno naznačiti broj osvojenih poena na domaćem zadatku u sledećem formatu:*

*Rok u kome je odbranjen domaći _____/(npr: jun 2007)
Ukupno poena _____ /40*

- *Na ispitnoj vežbanci potrebno je naznačiti zadatke koji su radjeni (zaokruživanjem broja zadatka).*
- *Nepoštovanje nekog od pravila povlači negativne poene.*

Pitanja

1. [8] Navesti pravila transformacije desno-linearne gramatike u regularnu.

Pogledati predavanja IR4PP1_P4 (Sintaksna analiza)

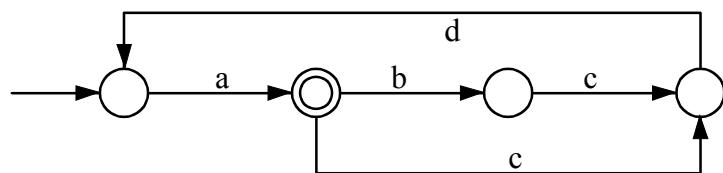
2. [8] Opšti algoritam rada potisnog automata.

Pogledati predavanja IR4PP1_P5 (Parsiranje od dna ka vrhu)

Zadaci

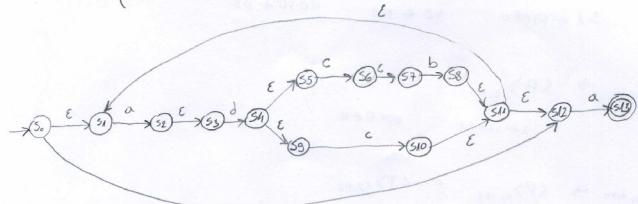
3. [9]

- a) [5] Napisati regularni izraz koji opisuje skup sekvenci koje automat sa slike prihvata.
- b) [4] Konstruisati minimalni deterministički konačni automat koji prihvata sve i samo one sekvence koje prihvata i automat sa slike, ali čitane s desna u levo.



a) $\alpha ((bc|c)d\alpha)^*$

b) $(\alpha d (cb|c))^* \alpha$



	a	b	c	d	
S ₀ , S ₁ S ₁₂	S ₂ , S ₃ , S ₁₃		/	/	∅ A
S ₂ , S ₃ S ₁₃		/		S ₄ , S ₅ , S ₉	1 B
S ₄ , S ₅ S ₉	/	/	S ₆ , S ₇ , S ₁₀ S ₁₁ , S ₁₂ , S ₁	/	∅ C
S ₆ , S ₇ S ₁₀ , S ₁₁ S ₁₂ , S ₁	S ₂ , S ₃ , S ₁₃ S ₁	S ₈ , S ₁₁ , S ₁₂ S ₁		/	∅ D
S ₈ , S ₁₁ S ₁₂ , S ₁	S ₂ , S ₃ , S ₁₃		/	/	∅ E

Nema suvišnih stanja.

Nakon minimizacije

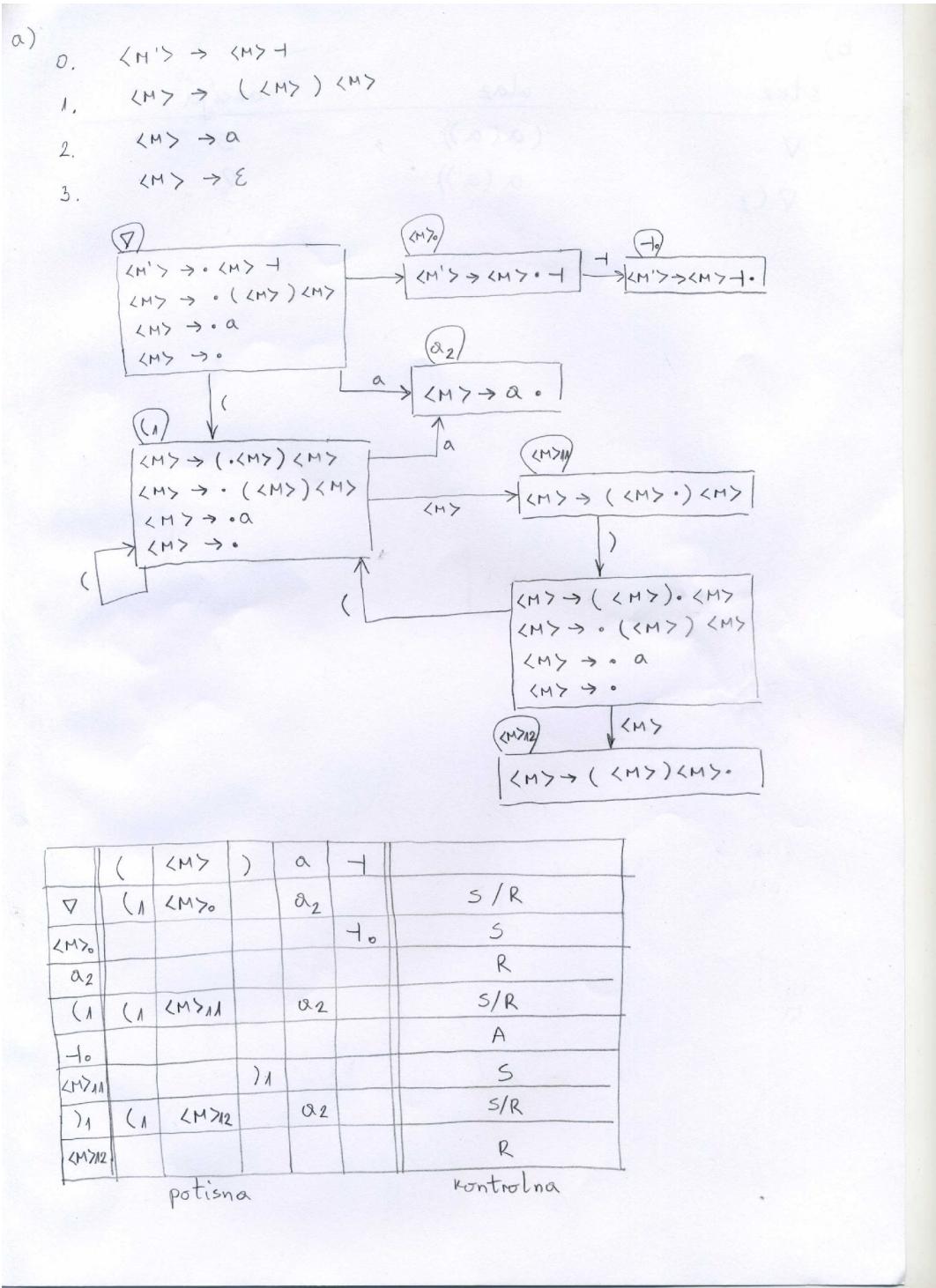
	a	b	c	d	
A	B			∅	
B			C	1	
C		D		∅	
D	B	E		∅	
E	B			∅	

	a	b	c	d	
A	B			∅	
B			C	1	
C	B	A		∅	
D	B			∅	

4. [8] Za datu gramatiku:

1. $\langle M \rangle \rightarrow (\langle M \rangle) \langle M \rangle$
2. $\langle M \rangle \rightarrow a$
3. $\langle M \rangle \rightarrow \epsilon$

- a) [4] Konstruisati LR(0) karakteristični automat po konfiguracionom modelu parsiranja. Prikazati izgled potisne i kontrolne tabele.
- b) [4] Prikazati izgled steka u svakom koraku prilikom parsiranja sekvene (a(a)).



b)

stek	ulaz	akcija
∇	$(\alpha(\alpha))$	S
$\nabla(\alpha)$	$\alpha(\alpha)$?

5. [7] Gramatika na slici sa startnim simbolom $\langle A \rangle$ opisuje regularne izraze sa operatorima unije $|$ i konkatenacije \circ .

- | |
|--|
| 1. $\langle A \rangle \rightarrow \langle Q \rangle$ |
| 2. $\langle Q \rangle \rightarrow \langle Q \rangle \langle P \rangle$ |
| 3. $\langle Q \rangle \rightarrow \langle P \rangle$ |
| 4. $\langle P \rangle \rightarrow \langle P \rangle \circ \langle T \rangle$ |
| 5. $\langle P \rangle \rightarrow \langle T \rangle$ |
| 6. $\langle T \rangle \rightarrow c$ |
| 7. $\langle T \rangle \rightarrow (\langle Q \rangle)$ |

Zadatoj gramatici dodati atribute i akcione simbole tako da se dobije atributivno-translaciona gramatika koja prevodi regularni izraz u opis nedeterminističkog konačnog automata koristeći atome:

STARTNOS(s) – označava da s predstavlja startno stanje automata

PRELAZ(s₁, s₂, a) – označava da iz stanja s₁ postoji prelaz u stanje s₂ pod ulazom a

PRIHVATAS(s) – označava da je s stanje prihvatanja

Primer: Izraz $c_a \circ c_b$ gde su a i b sintetizovani atributi terminala c, prevodi se u niz atoma STARTNOS(1); PRELAZ(1, 2, a); PRELAZ(2, 3, b); PRIHVATA(3).

Koristiti funkciju NOVOS koja pri svakom pozivu vraća novu oznaku stanja.

1. $\langle A \rangle \rightarrow \{ \text{STARTHOS} \}_{\text{START}} \quad \langle Q \rangle_{\text{STARO, NOVO}} \quad \langle \text{PRIHVATAS} \rangle_{\text{ciu}}$
 $(\text{START}, \text{STARO}) \leftarrow \text{NOVOS} \quad \text{CILJ} \leftarrow \text{NOVO}$

2. $\langle Q \rangle_{\text{STARO, NOVO}} \rightarrow \langle Q \rangle_{S_1, N_1} \quad | \quad \langle P \rangle_{S_2, N_2}$
 $S_1 \leftarrow \text{STARO} \quad S_2 \leftarrow N_1 \quad \text{NOVO} \leftarrow N_2$

3. $\langle Q \rangle_{\text{STARO, NOVO}} \rightarrow \langle P \rangle_{S, N}$
 $S \leftarrow \text{STARO} \quad \text{NOVO} \leftarrow N$

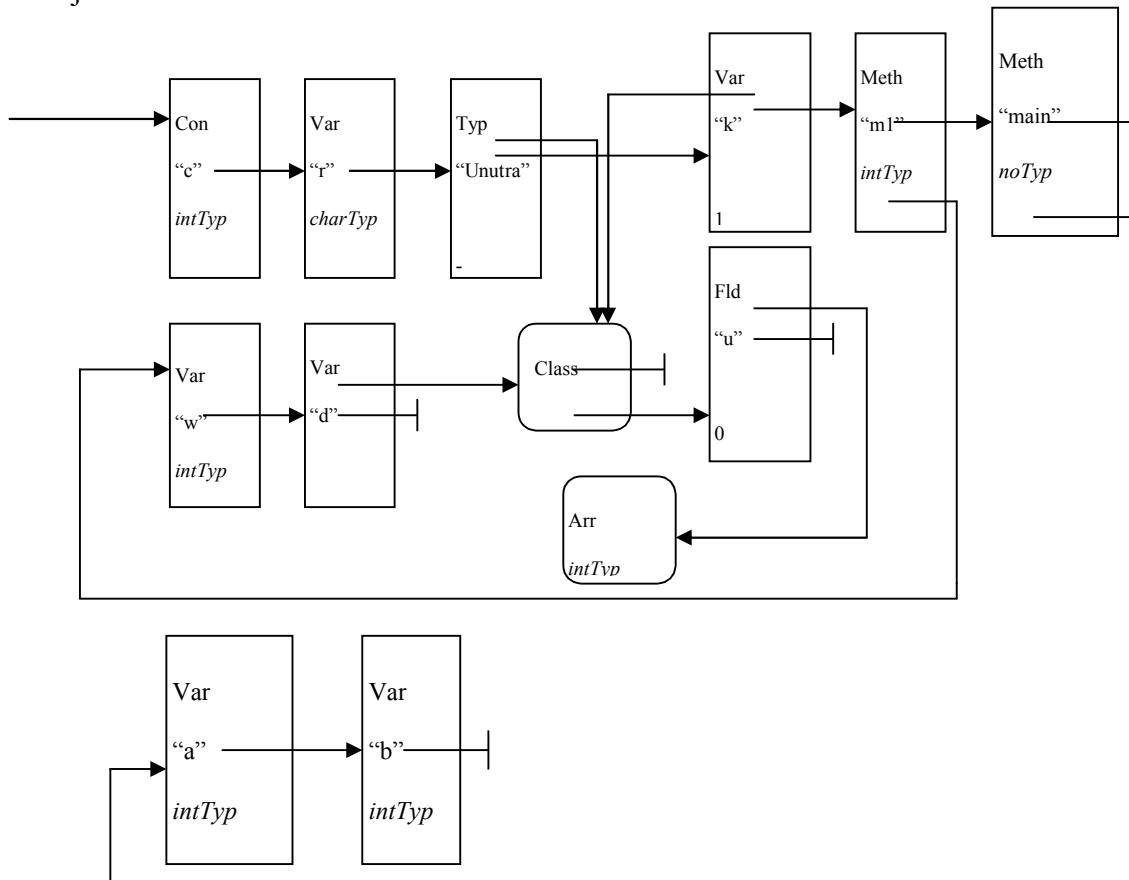
4. $\langle P \rangle_{\text{STARO, NOVO}} \rightarrow \langle P \rangle_{S_1, N_1} \quad | \quad \langle T \rangle_{S_2, N_2}$
 $S_1 \leftarrow \text{STARO} \quad S_2 \leftarrow N_1 \quad \text{NOVO} \leftarrow N_2$

5. $\langle P \rangle_{\text{STARO, NOVO}} \rightarrow \langle T \rangle_{S, N}$
 $S \leftarrow \text{STARO} \quad \text{NOVO} \leftarrow N$

6. $\langle T \rangle_{\text{STARO, NOVO}} \rightarrow C_X \quad \{ \text{PRELAZ} \}_{S, N, U}$
 $S \leftarrow \text{STARO} \quad U \leftarrow X \quad (\text{NOVO}, N) \leftarrow \text{NOVOS}$

7. $\langle T \rangle_{\text{STARO, NOVO}} \rightarrow (\langle Q \rangle_{S, N})$
 $S \leftarrow \text{STARO} \quad \text{NOVO} \leftarrow N$

6. [6] Na narednoj slici data je tabela simbola za neki deo programa pisanog na Mikro Java jeziku.



- a) [3] Na osnovu date tabele simbola rekonstruisati moguće delove Mirko Java koda.
b) [3] Da li će redosled deklaracija biti isti kao u orginalnom kodu? Objasniti.

a)

```

final int c = ...;
char r;
class Unutra {
    int u[];
}
Unutra k;

{
    int m1(?)
    ?
    {
    }

    void main(?)
    ?
    {
    }
}

```

Uz odgovarajuće, rekonstruisane, delove koda bilo je poželjno dati komentare koji opisuju na osnovu čega je dati deo rekonstruisan i zašto u navedenom obliku.

b) Redosled deklaracija biće istovetan kao u orginalnom kodu. Implementacija tabele simbola putem ulančane liste čuva redosled deklaracija.

7. [8] Za kod naveden u nastavku napisati troadresni kod, i njegovu implemntaciju putem četvorki i trojki.

```
*q = b[i-16];
b[i] = *q;
b[i+1] = a+3;
i++;
```

ako su promenljive i, a, q i b deklarisane na sledeći način: int i, a; int *q; int b[20];

Pogledati vežbe, oblast Generisanje medjukoda. Zadatak 4.

8. [6] Priložena tabela prikazuje neke od osnovnih instrukcija MJ virtuelne mašine. Upotrebom osnovnih instrukcija potrebno je prikazati i objasniti jednu moguću implementaciju sledećih složenih instrukcija:

- [3] **inc b** – inkrementira vrednost lokalne promenljive. **b** označava operand koji predstavlja redni broj lokalne promenljive u grupi deklaracija lokalnih promenljivih metode u čijem telu se nalazimo.
- [3] **incstatic s** – inkrementira vrednost globalne promenljive. **s** je redni broj globalne promenljive u grupi deklaracija globalnih promenljivih u glavnoj klasi.

Moguće je da neka od osnovnih instrukcija koja je potrebna za implementaciju nije navedena u priloženoj tabeli.

load b	... → ..., val
putstatic s	..., val → ...
getstatic s	... → ..., val
putfield s	..., adr, val → ...
const w	... → ..., val
add	..., val1, val2 → ..., val1+val2
new s	... → ..., adr
aload	..., adr, index → ..., val

a) i b)

load b const 1 add store b	getstatic s const 1 add putstatic s
-------------------------------------	--

Komentar uz kod je bio poželjan ali nije bio neophodan.