

---

Elektrotehnički fakultet u Beogradu

*Predmet:* Programski prevodioci 1

*Nastavnik:* doc. dr Dragan Bojić

*Ispitni rok:* Jun 2013.

*Datum:* 04.06.2013.

*Kandidat:* \_\_\_\_\_

*Broj Indeksa:* \_\_\_\_\_ *E-mail:* \_\_\_\_\_

*Ispit traje 3 sata. Nije dozvoljeno korišćenje literature.*

*Zadatak 1* \_\_\_\_\_/5

*Zadatak 5* \_\_\_\_\_/8

*Zadatak 2* \_\_\_\_\_/6

*Zadatak 6* \_\_\_\_\_/8

*Zadatak 3* \_\_\_\_\_/8

*Zadatak 7* \_\_\_\_\_/9

*Zadatak 4* \_\_\_\_\_/8

*Zadatak 8* \_\_\_\_\_/8

**Domaći zadatak:** \_\_\_\_\_/40

**Ocena:** \_\_\_\_\_

**Ukupno:** \_\_\_\_\_/60

**Napomena:** Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Na pitanja odgovarati **čitko i precizno**. Srećno!

---

1) (5 poena)

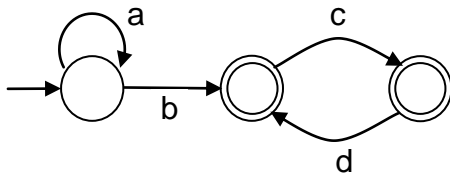
Staviti T u kućicu ako je iskaz tačan, F ako nije tačan.

Ispravni odgovori boduju se sa +1, neispravni sa -0.5, a ako se kućica ostavi prazna boduje se sa 0.

a) Za svaku bezkontekstnu gramatiku može se napisati ekvivalentan regularni izraz.	
b) Regularni izraz $(000)^+$ opisuje isključivo neprazne sekvence sa neparnim brojem nula.	
c) Moguće je parsirati indirektno levo rekurzivne gramatike.	
d) Vrednosti atributa terminala se nasleđuju iz roditeljskih čvorova stabla izvođenja.	
e) Sintaksom programskog jezika se opisuje, a gramatikom proverava, da li je svaka promenljiva deklarirana pre upotrebe.	

2) (6 poena)

Napisati regularni izraz ekvivalentan datom konačnom automatu:



**Rešenje:**

### 3) (8 poena)

Konstruisati automat koji obrađuje sekvence od jednog ili više sukcesivnih tagova. Format jednog taga je: <tagName:tagValue>. Između dva taga može postojati proizvoljno mnogo znakova razmaka. Identifikator tagName sastoji se od slova i cifara, pri čemu mora početi slovom.

Identifikator tagValue se sastoji od slova, cifara i znakova razmaka.

Ulazni simboli su "<", ":", ">", c, s, razmak, gde c označava cifru, a s označava slovo engleske abecede.

Primer sekvence: <name:Pera> <years:23><street:Programskih jezika 1>

- a) Napisati regularan izraz koji opisuje sekvence tagova.
- b) Tompsonovim algoritmom konstruisati nedeterministički automat za izraz iz tačke a) i konvertovati ga u minimalni deterministički konačni automat.

**Rešenje:**

#### 4) (8 poena)

Razmotriti sledeću gramatiku koja opisuje boolean izraze:

$S \rightarrow F$

$F \rightarrow F \wedge F \mid F \vee F \mid \neg F \mid (F) \mid id \mid true \mid false$

$F$  i  $S$  su neterminali,  $id$  je terminal koji označava boolean promenljivu, a  $true$  i  $false$  su terminali koji označavaju konstante.

a) Dodati atribut i akcije za izračunavanje vrednosti izraza. Radi jednostavnosti, pretpostaviti da sve promenljive imaju vrednost *true*.

b) Za literal (promenljivu ili konstantu) se kaže da ima pozitivnu fazu ako se u izrazu pojavljuje pod dejstvom parnog broja negacija, u suprotnom ima negativnu fazu.

Na primer, u izrazu  $\neg(a \vee \neg(b \wedge \neg c))$ ,  $a$  i  $c$  imaju negativnu, dok  $b$  ima pozitivnu fazu.

Dodati atribut i akcije za izračunavanje faze svakog literala u datoj gramatici (ne u onoj iz tačke a).

**Rešenje:**

**5) (8 poena)**

Posmatra se zadata gramatika.

1.  $\langle S \rangle \rightarrow \langle A \rangle \langle B \rangle$
2.  $\langle A \rangle \rightarrow \langle S \rangle a$
3.  $\langle A \rangle \rightarrow \varepsilon$
4.  $\langle B \rangle \rightarrow b$

- a) Konstruisati SLR(1) parser. Da li se radi o SLR(1) gramatici i zašto.
- b) Konstruisati top-down parser po principu rekurzivnog spusta za zadatu gramatiku.

**Rešenje:**

**6) (8 poena)**

Dat je sledeći program na jeziku sličnom Pascalu. Statičko okruženje za nelokalne promenljive je realizovano pomoću displeja. Displeji se čuvaju u memoriji.

a) Prikazati jasno i precizno izgled steka poziva neposredno pre povratka iz procedure c.

Voditi računa o formatu aktivacionih zapisa.

b) Napisati kompletan 80x86 asemblerski kod koji bi kompajler izgenerisao za procedure c i d (enter i leave instrukcije ne postoje).

Komentarisi svaku liniju koda.

**Rešenje:**

```
program Jun13 (output);
    var g, t: integer;

    procedure b(p: integer)
        var m: integer;

        procedure c ()
            begin
                g:=m+t;
            end; {c}
        procedure d (p:integer)
            t:=p+2;
            c();
        end {d}
    begin
        m:=p+1;
        d(m);
    end; {b}

begin
    g:= 1;
    b(g);
end. {ispit}
```

**7) (9 poena)**

Izgenerisati kompletan bajtkod za program Jun2013.

**Rešenje:**

```
program Jun2013
    const int K = 2;
    int p;
    class C {
        int d;
    {
        int m() { return K+d; }
    } }
    class DC extends C {
        int f;
    {
        int m() { return f-d; }
    } }
{
    void main()
        C c;
    {
        p = 3;
        c = new DC;
        c.d = K + p
        c.f = c.d + c.m();
    }
}
```

**8) (8 poena)**

Dat je sledeći fragment troadresnog međukoda:

t1 := - a

t2 := a \* t1

t3 := t2 - b

t4 := t2 \* t3

t5 := t4 + a

a := t5

Potrebno je uraditi sledeće:

- Za svaku instrukciju u kodu odrediti koje su promenljive žive.
- Korišćenjem algoritma *getreg* generisati mašinski kod za 80x86 arhitekturu na osnovu datog međukoda i informacija iz prethodne tačke c.

Za svaku instrukciju prikazati deskriptore registara, koji registar je vratila funkcija *getreg* i obrazloženje zašto je vraćen baš taj registar. Pretpostaviti da se koriste dvoadresne mašinske instrukcije gde je prvi operand određište operacije (oblika ADD dst, src); da se direktno memorijsko adresiranje može koristiti samo u MOV instrukciji i da se koriste dva registra AX i BX.

**Rešenje:**

	Ulaz	Život, sledeća upotreba.							Deskriptori		Generisani kod
		t1	t2	t3	t4	t5	a	b	AX	BX	
1	t1 := - a										
2	t2 := a * t1										
3	t3 := t2 - b										
4	t4 := t2 * t3										
5	t5 := t4 + a										
6	a := t5										