
Elektrotehnički fakultet u Beogradu

Predmet: Programski prevodioci 1
Nastavnik: dr Dragan Bojić, vanr. prof.
Ispitni rok: Septembar 2016.
Datum: 4.9.2016.

Kandidat: _____ *Broj Indeksa:* _____

Ispit traje 3 sata.
Nije dozvoljeno korišćenje literature
Prvih sat vremena nije dozvoljeno napuštati ispit.

<i>Zadatak 1</i>	_____ /6	<i>Zadatak 5</i>	_____ /8
<i>Zadatak 2</i>	_____ /6	<i>Zadatak 6</i>	_____ /8
<i>Zadatak 3</i>	_____ /8	<i>Zadatak 7</i>	_____ /8
<i>Zadatak 4</i>	_____ /8	<i>Zadatak 8</i>	_____ /8

Ispit: _____ /60

Ukupno: _____ /100

Projekat: _____ /40

Ocena: _____

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!

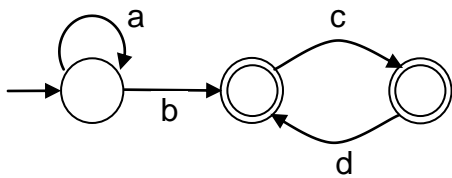
1) **(6 poena)**

Šta je poljska translaciona gramatika, zašto je ona važna i kako se neka translaciona gramatika transformiše u poljsku?

Rešenje:

2) **(6 poena)**

Napisati regularni izraz ekvivalentan datom konačnom automatu:



Rešenje:

3) (8 poena)

Neka je L_1 skup sekvenci koje prihvata prvi automat. Neka je N_2 skup sekvenci koje drugi automat N_2 prihvata. Konstruisati minimalni deterministički konačni automat koji prihvata sve sekvence iz skupa $L_1 \cup N_2$ (i nijednu više, gde je \cup operator unije).

	a	b	
→0	1		1
1	2	1	0
2		0	1

(prvi automat)

	b	c	
→A	B	C	0
B	C		1
C		A	0

(drugi automat)

Rešenje:

4) (8 poena)

1. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$	5. $\langle P \rangle \rightarrow (\langle E \rangle)$
2. $\langle E \rangle \rightarrow \langle T \rangle$	6. $\langle P \rangle \rightarrow \text{var}$
3. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$	7. $\langle P \rangle \rightarrow \text{const}$
4. $\langle T \rangle \rightarrow \langle P \rangle$	

a) Konstruisati karakterističan LR(0) automat, odrediti FOLLOW skupove i nacrtati kontrolnu SLR(1) tabelu parsera za datu gramatiku.

b) Dopuniti gramatiku atributima i akcionim simbolima tako da opisuje prevođenje izraza u postfiksnu formu u kojoj su sračunati konstantni podizrazi. Na primer, izraz $2 + 3 * 5 * x$ treba prevesti u $2 15 x * +$. Izraz $2 + 3 * y * 5$ se prevodi u $2 3 y * 5 * +$. Napomena: dozvoljene su i složenije konstrukcije (npr. if-then) u okviru akcija.

Rešenje:

5) (8 poena)

Posmatra se zadata gramatika.

1. $E \rightarrow E * B$
2. $E \rightarrow E + B$
3. $E \rightarrow B$
4. $B \rightarrow 0$
5. $B \rightarrow 1$

a) Prevesti datu gramatiku u LL(1) oblik. Jasno označiti sve transformacije i kratko prkomentarisati svaki korak.

b) Na osnovu gramatike pod a), implementirati top-down parser po principu rekurzivnog spusta (u pseudokodu).

Rešenje:

6) (8 poena)

Dat je sledeći program na jeziku sličnom Pascalu. Glavni program ima svoj aktivacioni zapis. Enter i leave instrukcije ne postoje. Povratna vrednost funkcije se ostavlja u registru AX.

- a) Napisati 80x86 asemblerski kod, koji bi kompajler izgenerisao za proceduru *c*, ako je statičko okruženje za nelokalne promenljive realizovano pomoću pristupnih veza.

Rešenje:

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

  function a(k: integer):integer;
  begin
    return k + g;
  end; {a}
  procedure b(p: integer);
  var m: integer;

  procedure c(k:integer);
  begin
    d(a(p+g)+k);
  end; {c}

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

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

7) (8 poena)

Napisati kompletan bajtkod SAMO ZA funkciju *main* na osnovu priloženog listinga programa napisanog na jeziku Mikrojava. Sve metode unutrašnjih klasa su virtuelne. Globalne metode se pozivaju statički.

Rešenje:

```
program Mikrojava
  const int K = 2;
  class C {
    int d;
  {
    int m(int c)
    { return c * d++; }
  } }
  class DC extends C {
    int v;
  {
    int m(int q)
    { return d + q; }
  } }
{
  void main()
  C c[];
  {
    c = new C[1];
    c[0] = new DC;
    new C().m(c[0].m(3));
  }
}
```

8) (8 poena)

Dat je sledeći fragment troadresnog međukoda:

```

a := 3
b := a * 4
t1 := b - a
t2 := t1 + b
a := a * t2

```

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.

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. Promenljive a, b su korisničke, dok su t1 i t2 privremene.

Rešenje:

	Ulaz	Život, sledeća upotreba				Deskriptori		Generisani kod
		t1	t2	a	b	AX	BX	
1	a := 3							X86
2	b := a * 4							
3	t1 := b - a							
4	t2 := t1 + b							
5	a := a * t2							