Elektrotehnički fakultet u Beogradu

Predmet: Programski prevodioci 1
Nastavnik: doc. dr Dragan Bojić
Ispitni rok: Jun 2015.
Datum: 09.06.2015.

Kandidat: _____ *Broj Indeksa:* _____

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

Zadatak 1	_____/6	*Zadatak 5*	_____/8
Zadatak 2	_____/6	*Zadatak 6*	_____/8
Zadatak 3	_____/8	*Zadatak 7*	_____/8
Zadatak 4	_____/8	*Zadatak 8*	_____/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)**

Na programskom jeziku Java napisati implementaciju Mikrojava instrukcije *astore*. Postoje sledeći memorijski segmenti, kojima se može pristupiti kao jednodimenzionalnim nizovima preko sledećih identifikatora: *estack* (stek izraza), *heap* (dinamička memorija). Operacija *pop* skida element sa vrha steka, a operacija *push* stavlja vrednost na vrh steka.

Rešenje:

2) **(6 poena)**

Ulazni alfabet čine slova {a, b} i zagrade ( ). Potrebno je napisati gramatiku za sekvence kod kojih su istovremeno zadovoljeni sledeći uslovi:

• iza svakog a odmah dolazi (bar jedno) b

• zagrade su uparene.

Na primer, ispravna sekvenca je ab((abbab)( )) a neispravne su a(b)abb, ab)ab(babab.

Rešenje:

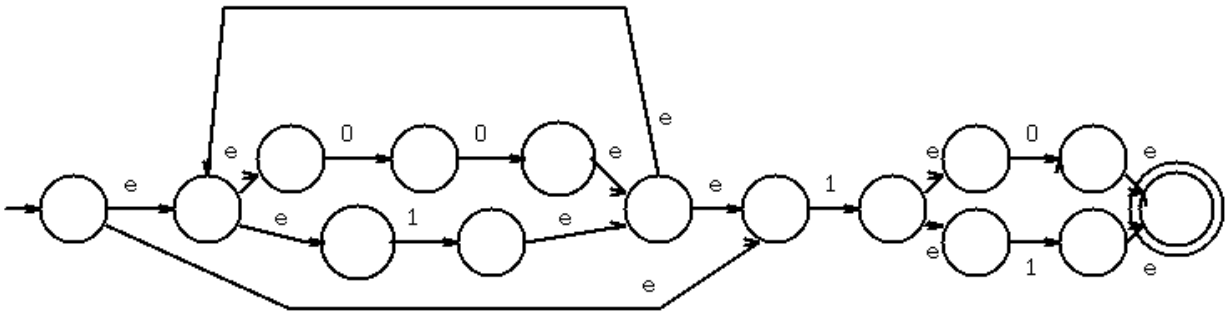
3) **(8 poena)**

a) Za automat na slici (e predstavlja praznu sekvencu) konstruisati ekvivalentan DKA.

b) Za polazni automat (na slici) odrediti ekvivalentni regularni izraz.

c) Za regularni izraz pod b) konstruisati sintaksno stablo i odrediti funkcije prva, poslednja i sledeća pozicija. Na osnovu ovih funkcija konstruisati DKA.

d) Proveriti da li su automati dobijeni pod tačkama a) i c) ekvivalentni. Ako nisu, navesti sekvencu razlikovanja.



Rešenje:

4) **(8 poena)**

Data je gramatika za jednostavan programski jezik:

1. ⟨program⟩ → ⟨niz_naredbi⟩
2. ⟨niz naredbi⟩ → ⟨naredba⟩ ⟨niz_naredbi⟩
3. ⟨niz naredbi⟩ → ε
4. ⟨naredba⟩ → P = C
5. ⟨naredba⟩ → IF P THEN ⟨niz naredbi⟩ ENDIF
6. ⟨naredba⟩ → WHILE P ⟨niz naredbi⟩ ENDWHILE

Želi se napraviti "pretty printer", tj. alat koji ispisuje listing ulaznog programa sa po jednom naredbom u liniji, sa pravilnim nazubljivanjem, na primer:

```
WHILE P
   P = C
   IF P THEN
      P = C
   ENDIF
ENDWHILE
```

a) Dodati u gramatiku atribute i akcije da se postigne traženi ispis programa. Gramatika treba da je L atributivna. Od akcija je dozvoljeno koristiti isključivo sledeče: PRT(s) koja na izlaz ispisuje dati string s (u s nisu dozvoljeni spec. znaci, samo slova, cifre i znak =), TAB(n) koja na izlazu ispisuje n tabulatora i CRLF koja ispisuje jedan znak za novi red. Pretpostaviti da terminal P ima string atribut sa imenom date promenljive, a terminal C ima string atribut sa vrednošću konstante, npr. “123”. U atributivnim pravilima nije dozvoljena nikakva manipulacija stringovima, osim eventualno njihovog nadovezivanja s1+s2.

b) Odrediti SELECT skupove smena i konstruisati parser na bazi rekurzivnog spusta za atributivnu gramatiku iz tačke a). Dovoljno je napisati glavni program i deo parsera za smene 1, 2, 3 i 4.

Rešenje:

5) **(8 poena)**

Data je gramatika (znak | se koristi za skraćeno pisanje više smena istog neterminala):

$\langle S \rangle \rightarrow \langle A \rangle \langle S \rangle \mid b$

$\langle A \rangle \rightarrow \langle S \rangle \langle A \rangle \mid a$

a) Pokazati da je gramatika dvosmislena.

b) Konstruisati SLR(1) parser i navesti eventualne konflikte.

c) Konstruisati LALR(1) parser i naveti eventualne konflikte.

Rešenje:

6) **(8 poena)**

Dat je sledeći program na jeziku sličnom Pascalu. Glavni program ima svoj aktivacioni zapis. Enter i leave instrukcije ne postoje.

a) Prikazati izgleda steka aktivacionih zapisa neposredno pre povratka iz procedure d.
b) Napisati 80x86 asemblerski kod, koji bi kompajler izgenerisao za proceduru *c*, ako je statičko okruženje za nelokalne promenljive realizovano pomoću pristupnih veza.

Rešenje:

```
program Jun15 (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(t+k));
       end; {c}
       procedure d(p:integer);
       begin
             g := t + p;
       end; {d}
   begin
       m := p+1;
       c(m);
   end; {b}

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

7) **(8 poena)**

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

Rešenje:

```
program Jun2015
   const int K = 2;
   int p[];
   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;
   {
       p = new int[2];
       c = new DC;
       c.d = K;
       p[1] = c.m(3);
       p[1]--;
   }
}
```

8) **(8 poena)**

Dat je sledeći fragment troadresnog međukoda:

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

Potrebno je uraditi sledeće:

a) Za svaku instrukciju u kodu odrediti koje su promenljive žive.
b) 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 odredište 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	X86
1	a := 3							
2	b := a * 4							
3	t1 := b - a							
4	t2 := t1 + b							
5	a := t1 * t2							