
Elektrotehnički fakultet u Beogradu

Predmet: Programski prevodioci 1

Nastavnik: doc. dr Dragan Bojić

Ispitni rok: Jul 2013.

Datum: 25.06.2013.

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 _____/9

Zadatak 7 _____/8

Zadatak 4 _____/8

Zadatak 8 _____/8

Domaći zadatak: _____/40

Ispit: _____/60

Ocena: _____

Ukupno: _____/100

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) Kompajler radi sve provere da li se funkcija koja vraća ceo broj uvek završava sa return.	
b) Panični oporavak od greške u nekim situacijama dodaje sigurne simbole u ulazni niz.	
c) Ako parser završi rad sa ACCEPT, to znači da je ulaz semantički ispravan.	
d) Ako se razlikuje redosled smena u krajnje levom i krajnje desnom izvođenju iste ulazne sekvence, to znači da je gramatika dvosmislena.	
e) Virtuelna mašina za programske jezike uvek se realizuje kao stek mašina.	

2) (6 poena)

Razmotriti sledeću flex specifikaciju:

```
c          { printf("1"); }
ac+b*     { printf("2"); }
cc        { printf("3"); }
ab*       { printf("4"); }
a         { printf("5"); }
ac*b+     { printf("6"); }
```

Koja flex pravila ne mogu nikad biti primenjena i zašto?

Rešenje:

3) (9 poena)

- a) Za regularni izraz $(a*b)+c*a$ konstruisati deterministički konačni automat metodom pozicija. Automat prikazati grafički i tabelarno.
- b) Odrediti konačni automat koji prihvata iste sekvence kao i automat u tački a) ali u obrnutom redosledu.
- c) Particionom metodom proveriti da li su konačni automati u tačkama a) i b) ekvivalentni.

Rešenje:

4) (8 poena)

Na papiru na kome je neko napisao gramatiku postoji mrlja, pa se ne vide dve smene, jedna za S, druga za A. Sva sreća dati su FIRST i FOLLOW skupovi neterminala. Potrebno je odrediti nedostajuće dve smene (zabranjeno je koristiti uspravnu crtu, jer ako se napiše npr. $P \rightarrow Q \mid R$, to su u stvari dve smene $P \rightarrow Q$ i $P \rightarrow R$). $\$$ je marker kraja ulazne sekvence. Obrazložiti odgovor.

	FIRST	FOLLOW
S	{a,b,c,d,e}	{b,d,e,\$}
A	{c,d,e}	{b}
X	{c}	{d,e}
Y	{d}	{e}
Z	{e}	{b}

1. $S \rightarrow a S$
2. $S \rightarrow \dots ???$
3. $A \rightarrow \dots ???$
4. $A \rightarrow \varepsilon$
5. $X \rightarrow c S$
6. $X \rightarrow \varepsilon$
7. $Y \rightarrow d S$
8. $Y \rightarrow \varepsilon$
9. $Z \rightarrow e S$

Rešenje:

5) (8 poena)

Dat je fragment jedne gramatike kojom su opisani aritmetički izrazi u nekom programskom jeziku.

$E \rightarrow E * E$ // množenje

$E \rightarrow E + E$ // sabiranje

$E \rightarrow (E)$ // grupisanje zagrada

$E \rightarrow id$ // simbolička konstanta

Zadatu gramatiku je potrebno modifikovati tako da se podrže prioriteta operacija poređani od najvišeg ka najnižem: zagrade, množenje, sabiranje. Pored toga, množenje i sabiranje treba da budu levo asocijativne operacije.

- Korigovati datu gramatiku tako da se podrže navedena ograničenja i prevesti je u LL(1) oblik.
- Dodati attribute terminalima i neterminalima u smenama dobijenim pod a), tako da se omogući računanje vrednosti izraza.
- Konstruisati parser od vrha ka dnu na osnovu dobijene atributivne gramatike pod b).

Rešenje:

6) (8 poena)

Dat je sledeći program na jeziku sličnom Pascalu. Statičko okruženje za nelokalne promenljive je realizovano pomoću pristupnih veza. Glavni program poseduje sopstveni aktivacioni zapis na steku.

- 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) (8 poena)

Izgenerisati kompletan bajtkod za program Jun2013.

Rešenje:

```
program Jun2013
    const int K = 2;
    int p;
    class C {
        int d;
    {
        int m() { return p+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 * K;
        c.m();
    }
}
```

8) (8 poena)

Na programskom jeziku Java implementirati sledeće instrukcije Mikrojava bajtkoda koristeći elemente arhitekture Mikrojava virtuelne mašine objašnjenje u materijalima sa predavanja i vežbi.

- a) enter b1, b2
b1 označava broj argumenata funkcije, a
b2 predstavlja zbir broja argumenata i lokalnih promenljivih.
- b) putfield s
s prestavlja redni broj polja unutar objekta u koje treba upisati zadatu vrednost.

Rešenje: