
Elektrotehnički fakultet u Beogradu

Predmet: Programski prevodioci 1
Nastavnik: dr Dragan Bojić, vanr. prof.
Asistenti: mast.inž. Maja Vukasović
mast.inž. Kristijan Žiža
Ispitni rok: Jul 2020.
Datum: 26.06.2020.

Kandidat: _____ *Broj Indeksa:* _____

Ispit traje 150 minuta.
Nije dozvoljeno korišćenje literature.
Prvih sat vremena nije dozvoljeno napuštati ispit.

Zadatak 1 _____/10

Zadatak 4 _____/10

Zadatak 2 _____/10

Zadatak 5 _____/10

Zadatak 3 _____/10

Zadatak 6 _____/10

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) (10 poena)

Data gramatika predstavlja specifikaciju programskog segmenta koji opisuje metodu. Cilj zadatka je prebrojati ukupan broj različitih mogućih putanja u *control flow* grafu opisane metode.

Dodeliti attribute datoj gramatici tako startni neterminal MethodBody ima sintetizovani atribut koji predstavlja ukupan broj različitih putanja u telu metode. Smatrati da neterminali DesignatorStatement, Expression i Condition ne sadrže nikakva grananja. Za svaki atribut odrediti i njegov tip.

1. $\langle \text{MethodBody} \rangle \rightarrow \{ \{ \langle \text{Statement} \rangle \} \}$
2. $\langle \text{Statement} \rangle \rightarrow \langle \text{DesignatorStatement} \rangle ;$
3. $\langle \text{Statement} \rangle \rightarrow \text{"if" "(" } \langle \text{Condition} \rangle \text{ ")" } \langle \text{Statement} \rangle \text{ ["else" } \langle \text{Statement} \rangle \text{]}$
4. $\langle \text{Statement} \rangle \rightarrow \text{"return" } [\langle \text{Expression} \rangle] ;$
5. $\langle \text{Statement} \rangle \rightarrow \{ \{ \langle \text{Statement} \rangle \} \}$

Rešenje:

$\langle \text{MethodBody} \rangle \rightarrow \{ \{ \langle \text{Statement} \rangle \} \}$

$\langle \text{Statement} \rangle \rightarrow \langle \text{DesignatorStatement} \rangle ;$

$\langle \text{Statement} \rangle \rightarrow \text{"if" "(" } \langle \text{Condition} \rangle \text{ ")" } \langle \text{Statement} \rangle \text{ ["else" } \langle \text{Statement} \rangle \text{]}$

$\langle \text{Statement} \rangle \rightarrow \text{"return" } [\langle \text{Expr} \rangle] ;$

$\langle \text{Statement} \rangle \rightarrow \{ \{ \langle \text{Statement} \rangle \} \}$

2) (10 poena)

Neka se sekvence na ulazu nekog automata sastoje isključivo od malih “()” i uglastih zagrada “[]”.

Konstruisati potisni automat koji proverava da li su zagrade u toj sekvenci ispravno uparene.

Primeri ispravnih izraza: [[([)]])(), (()), []() .

Primeri neispravnih izraza [], [()], [)] .

Rešenje:

3) (10 poena)

Zadata je sledeća atributivno-translaciona gramatika:

a) Odrediti SELECT skupove.

b) Napisati procesor na bazi rekurzivnog spusta za datu atributivno-translacionu gramatiku.

1. $\langle S \rangle_s \rightarrow \langle A \rangle_{i1,s1} c_x \langle A \rangle_{i2,s2}$
 $i1 \leftarrow 5 \quad i2 \leftarrow i1 * s1 + x \quad s \leftarrow s2$
2. $\langle S \rangle_s \rightarrow a_x \{D\}_i$
 $s, i \leftarrow x$
3. $\langle A \rangle_{i,s} \rightarrow \varepsilon$
 $s \leftarrow i$
4. $\langle A \rangle_{i,s} \rightarrow b_x \langle S \rangle_{s1}$
 $s \leftarrow x + s1$

Rešenje:

4) (10 poena)

a) Podeliti dole dati segment koda u bazične blokove i nacrtati graf toka kontrole na nivou bazičnih blokova.

```
L1: x:=x+1
L2: x:=x+1
L3: x:=x+1
    if x=1 then goto L1
    if x=2 then goto L3
    if x=3 then goto L5
L4: x:=x+1
L5: x:=x+1
    if x=4 then goto L4
```

b) Napisati troadresni kod, i njegovu implementaciju putem četvorki i trojki, koji odgovara sledećoj sekvenci na C-u:

```
if (a <= b) {
    funkcija(3, c);
    a++;
}
b++;
```

Rešenje:

5) (10 poena)

Prikazani su izgled steka i operacije koje izvršava „potisni-svedi“ parser prilikom procesiranja niza znakova *aadccc* i *abd*. Odrediti gramatiku na osnovu koje je formiran ovaj parser. Odrediti FOLLOW skupove svih neterminala dobijene gramatike i odgovoriti da li gramatika pripada skupu SLR(1) gramatika.

Stek	Ulazni niz	Operacija
∇	aadccc┐	SHIFT
∇a	adccc┐	SHIFT
∇aa	dccc┐	SHIFT
∇aad	ccc┐	SHIFT
∇aadc	cc┐	REDUCE
∇aad	cc┐	REDUCE
∇aa<S>	cc┐	REDUCE
∇a<A>	cc┐	SHIFT
∇a<A>c	c┐	SHIFT
∇a<A>cc	┐	REDUCE
∇a<A>c	┐	REDUCE
∇a<A>	┐	REDUCE
∇<S>	┐	ACCEPT

Stek	Ulazni niz	Operacija
∇	abd┐	SHIFT
∇a	bd┐	SHIFT
∇ab	d┐	REDUCE
∇a<A>	d┐	SHIFT
∇a<A>d	┐	REDUCE
∇a<A>	┐	REDUCE
∇<S>	┐	ACCEPT

Rešenje:

Gramatika pripada skupu SLR(1): DA NE

6) (10 poena)

Prevođenjem Mikrojava programa dobijen je sledeći Mikrojava bajtkod. Rekonstruisati izgled programa koji je preveden. Nazive identifikatora izabrati proizvoljno. Ukoliko je dati bajtkod moguće dobiti iz više različitih Mikrojava programa izabrati i prikazati jedno moguće rešenje. U rešenju jasno naznačiti koji deo datog bajtkoda se odnosi na koju liniju programa.

Rešenje:

```
enter 0 2
const 5
putstatic 0
getstatic 0
neg
const 7
add
putstatic 1
getstatic 1
newarray 0
store_0
load_0
const_0
getstatic 0
astore
load_0
const_1
load_0
const_0
aload
const_2
sub
astore
new 4
store_1
load_1
load_0
const_1
aload
putfield 0
exit
return
```