
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programski Prevodioci 1 (IR4PP1)
Nastavnik: doc dr Dragan Bojić
Asistent: Dipl. Ing. Miloš Gligorić
Ispitni rok: April 2008.
Datum: 11.05.2008.

Napomene:

- *Ispit traje 3 sata. Prvih 30 minuta nije dozvoljeno napuštanje sale. Prva dva sata nije dozvoljeno iznositi ispitne zadatke. Poslednjih 10 minuta ispita nije dozvoljeno napuštanje sale.*
- *Na ispitnoj vežbanci obavezno naznačiti broj osvojenih poena na domaćem zadatku u sledećem formatu:*

Rok u kome je odbranjen domaći _____/(npr: jun 2007)
Ukupno poena _____/40

- *Na ispitnoj vežbanci potrebno je naznačiti zadatke koji su radjeni (zaokruživanjem broja zadatka).*
- *Nepoštovanje nekog od pravila povlači negativne poene.*

Pitanja

1. [8] Navesti osnovna pravila konzistencije atributivne gramatike
2. [8] Navesti formulu za adresiranje elementa $A[i_1, i_2]$ dvodimenzionalne matrice A smeštene u memoriju po vrstama, gde je w memorijska veličina jednog elementa matrice u adresibilnim jedinicama, low_i je najmanji indeks po i toj dimenziji, a $high_i$ najveći, dok je $base$ početna adresa matrice A .

Zadaci

3. [4] Napisati regularne izraze za slučajeve za koje je to moguće. Detaljno objasniti.
- [1] String sa azbukom {a, b, c} gde prvo a prethodi prvom b.
 - [1] String sa azbukom {a, b, c} gde je sigurno paran broj karaktera a.
 - [1] String sa azbukom {a, b} gde se a javlja više puta od b.
 - [1] String sa azbukom {a, b, c} u kome se ne pojavljuje podstring baa.

4. [9] Data je gramatika:

1. $\langle R \rangle \rightarrow 7 \langle R \rangle 3$
2. $\langle R \rangle \rightarrow 3 \langle R \rangle c$
3. $\langle R \rangle \rightarrow 3 c$

- [6] Upotrebom konfiguracione metode konstruisati SLR(1) parser koji odgovara gramatici u postavci zadatka. Nacrtati karakteristični automat, potisnu i kontrolnu tabelu.
- [3] Prikazati izgled steka iz prethodne tačke, po koracima, pri parsiranju ulaza 73c3c. U svakom koraku prikazati izgled steka, preostali ulaz i akciju koja se vrši.

5. [7] Data je sledeća gramatika

1. $\langle P \rangle \rightarrow \langle P \rangle q \langle M \rangle$	5. $\langle B \rangle \rightarrow d$
2. $\langle P \rangle \rightarrow q \langle D \rangle$	6. $\langle D \rangle \rightarrow \langle D \rangle a \langle M \rangle$
3. $\langle P \rangle \rightarrow \langle D \rangle$	7. $\langle D \rangle \rightarrow \langle B \rangle b \langle M \rangle$
4. $\langle M \rangle \rightarrow b \langle M \rangle$	8. $\langle D \rangle \rightarrow \langle M \rangle$

- [4] Iz date gramatike eliminisati levu rekurziju.
- [3] Izvršiti levu faktorizaciju u gramatici dobijenoj pod a).

6. [7] Neka je jedan iskaz medjukoda opisan sledećom klasom (Java):

```
public abstract class Iskaz {  
  
    /* true ako je iskaz uslovni skok */  
    public abstract boolean uslovniSkok();  
  
    /* true ako je iskaz bezuslovni skok */  
    public abstract boolean bezuslovniSkok();  
  
    /* vraca odrediste skoka za skokove, inace  
     * vrednost nije definisana  
     */  
    public abstract int odredisteSkoka();  
}
```

i neka je bazični blok opisan sledećom klasom:

```

public class Blok {
    /* dodaje iskaze u blok redom */
    public void dodajIskaz(Iskaz i) {
        /* ... */
    }
}

```

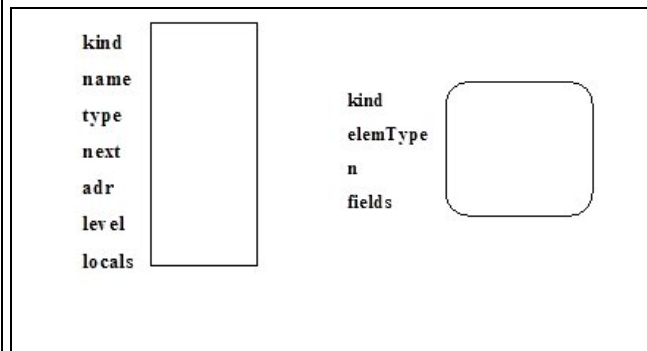
Napisati metodu koja kao argument prihvata niz Iskaza (Iskaz iskazi[]), a kao rezultat vraća niz bazičnih blokova određenih na osnovu dobijenog niza iskaza. Kreirati samo neophodan broj bazičnih blokova. Bazične blokove nije potrebno međusobno povezivati. Detaljno komentarisati kod.

7. [8] Za dati MikroJava program, nacrtati izgled tabele simbola u tački (* 6 *). Za umetanje umesto jednostruko ulančane liste koristiti binarno stablo (kao ključ se koristi ime; posmatra se leksikografski poredak). Znači i dalje postoje Object, Struct i Scope čvorovi (izgled ovih čvorova za slučaj liste je dat kao podsetnik). Universe opseg nije potrebno crtati.

```

class A
  int b;
  final int aa = 1;
  class Zz {
    int m;
    int ra[];
  }
  Zz z;
{
  int sum(char i)
  {
    x = 5 + y;
    y++; a += 3;
  }
  void ab()
  int rch, ss3 ;
  {
    rch += 12 ; ss3-- ;
    (* 6 *)}
}

```



8. [9] Napisati tri primera, uz detaljno objašnjenje, na programskom jeziku Pascal (pseudo Pascal), koja prikazuju tri moguća načina formiranja pristupne veze. Nacrtati izgled steka za date primere, koji ilustruje kreiranje pristupne veze. Primeri trebaju da budu jednostavni, a svaki deo primera treba ukratko prokomentarisati.