
Elektrotehnički fakultet u Beogradu

Predmet: Programski prevodioci 1

Nastavnik: doc. dr Dragan Bojić

Ispitni rok: Januar 2013.

Datum: 20.01.2013.

Kandidat: _____

Broj Indeksa: _____ *E-mail:* _____

Ispit traje 3 sata. Nije dozvoljeno korišćenje literature.

Zadatak 1 _____/6

Zadatak 5 _____/8

Zadatak 2 _____/8

Zadatak 6 _____/8

Zadatak 3 _____/8

Zadatak 7 _____/8

Zadatak 4 _____/8

Zadatak 8 _____/6

Domaći zadatak: _____/40

Ocena: _____

Ukupno: _____/60

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)

Za svaku od sledećih aktivnosti odgovoriti da li nastaje u vreme pisanja kompajlera (DESIGN), u vreme prevođenja korisničkog programa kompajlerom (COMPILE), ili u vreme izvršavanja programa (RUN):

- a) Određivanje pomeraja lokalne promenljive C programa
- b) Određivanje implementacije (strukture podataka) za troadresni kod.
- c) Određivanje adrese aktivacionog zapisa procedure.
- d) Određivanje koliko pristupnih veza treba slediti da bi se došlo do nelokalne Pascal promenljive.
- e) Određivanje redosleda smeštanja na stek stvarnih parametara.
- f) Sadržaj displeja na leksičkom nivou 3.

Rešenje:

2) (8 poena)

- a) Napisati regularan izraz koji opisuje aritmetičke izraze sa nenegativnim celobrojnim operandima i operacijama +, -, * i / bez zagrada (na primer, $0015+35*50$). Ulazni simboli su +, -, *, /, c gde c označava cifru (najkraći izraz je samo broj).
- b) Tompsonovim algoritmom konstruisati nedeterministički automat za izraz iz tačke a) i konvertovati ga u minimalni deterministički konačni automat.
- c) Konvertovati automat pod b), u procesor koji računa vrednost aritmetičkog izraza. Prioritet svih operacija je isti a redosled računanja je s leva na desno, pa je na primer $15+35*50 = 2500$.

Rešenje:

3) (8 poena)

- Za sledeću gramatiku konstruisati karakterističan LR(0) automat (bez parserskih tabela).
- Da li je gramatika LR(0)? Obrazložiti odgovor.
- Izračunati poništive neterminale, first i follow skupove neterminala i odgovoriti na pitanje da li je gramatika SLR(1) uz obrazloženje.
- Konstruisati potisnu i kontrolnu tabelu SLR(1) parsera (bez obzira na odgovor pod c).

Napomena: tretirati ** kao jedan znak. \$ je marker kraja ulaza.

| 0. $\langle E \rangle' \rightarrow \langle E \rangle \$$ | | First | follow | Poništiv |
|--|----------------------|-------|--------|----------|
| 1. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$ | $\langle E' \rangle$ | | | |
| 2. $\langle E \rangle \rightarrow \langle T \rangle$ | $\langle E \rangle$ | | | |
| 3. $\langle T \rangle \rightarrow x ** \langle T \rangle$ | $\langle T \rangle$ | | | |
| 4. $\langle T \rangle \rightarrow x$ | | | | |

Rešenje:

4) (8 poena)

Pretpostavimo da u iskaze (statement) nekog programskog jezika želimo dodati uslovni iskaz sledećeg oblika:

```
ifequal(exp1, exp2)
statement1
smaller
statement2
larger
statement3
```

Ako se pri izvršavanju ustanovi jednakost celobrojnih izraza $exp1$ i $exp2$, izvršiće se *statement1*, ako je $exp1 < exp2$ izvršiće se *statement2*, a ako je $exp1 > exp2$, izvršiće se *statement3*.

- Napisati gramatiku GI za ovaj iskaz tako da se delovi *smaller* i *larger* (jedan ili oba) mogu izostaviti. Ako su oba prisutna, onda treba da se pojave u navedenom redosledu.
- Koje provere treba da se ugrade u sintaksno-semantički analizator u vezi *ifequal* iskaza?
- Treba konstruisati atributivnu gramatiku AG koja opisuje proizvoljnu gramatiku, kao skup jedne ili više smena, koristeći skup terminala:

T (označava terminal), NT (neterminal), SEP (opisuje \rightarrow) i EPS (opisuje ϵ).

U gramatiku AG dodati attribute koji računaju za svaki neterminal da li je poništiv, ako svaki neterminal ima atribut *i* koji je njegov jedinstveni indeks, a postoji globalni vektor PONIŠTIV čiji element *i* treba na kraju da daje informaciju o poništivosti *i-tog* neterminala. Smene ne moraju biti grupisane prema levoj strani smene.

- Da li je moguće izračunati poništivost neterminala jednim prolaskom kroz stablo izvođenja gramatike AG? Ako jeste, obrazložiti kakvom vrstom prolaska kroz stablo. Ako nije, obrazložiti zašto.

Rešenje:

5) (8 poena)

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

1) Prikazati jasno i precizno izgled aktivacionog zapisa samo za poziv procedure d. Voditi računa o formatu aktivacionog zapisa. Prikazati jasno deo koji je formirao pozivalac, a koji pozvana procedura. Poznate su sledeće adrese skzivacionih zapisa: &AZJan2013, &AZb, &AZd.

2) Napisati kompletan 80x86 asemblerski kod koji bi kompajler izgenerisao za procedure **c** i **b** (enter i leave instrukcije ne postoje). Komentarisati svaku liniju koda.

Rešenje:

```
program Jan2013 (output);
  var g: integer;

  procedure b(p: integer);
    var m: integer;

    procedure c (a: integer);
    begin
      m:=g+a;
    end; {c}

    procedure d (a: integer);
    begin
      c (a+3);
    end; {d}
  begin
    m := 2;
    d (g);
  end; {b}
begin
  g := 1;
  b(3);
end. {ispit}
```

6) (8 poena)

Data je sledeća sekvenca međukoda:

$t = x + 1$

$x = y + 3$

$z = x * t$

$y = y + 3$

$x = t * y$

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 tačke a. Za svaku instrukciju prikazati deskriptore registara, koji registar je vratila funkcija *getreg* i obrazloženje zašto je vraćen baš taj registar.

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`.

Rešenje:

7) (8 poena)

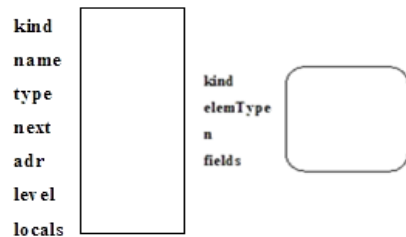
Izgenerisati kompletan bajtkod za program Jan2013.

Rešenje:

```
program Jan2013
    const int K = 2;
    int p;
    class C {
        int d;
    {
        int m(int a) {
            return a+d;
        }
    } }
{
    void main()
        C c;
    {
        p = 3;
        c = new C;
        c.d = K + p
        p = c.m(p);
    }
}
```


8) (6 poena)

- a) Nacrtati izgled/stanje tabele simbola na kraju parsiranja Mikrojava programa iz zadatka 7. Nije potrebno prikazivati (ceo) *universe* opseg važenja imena.



Rešenje: