
Elektrotehnički fakultet u Beogradu

Predmet: Programski prevodioci 1
Nastavnik: dr Dragan Bojić, red. prof.
Asistenti: mast.inž. Maja Vukasović
mast.inž. Mihajlo Ogrizović
Ispitni rok: Avgust 2023.
Datum: 26.08.2023.

Potpis dežurnog

Kandidat: _____

Broj Indeksa: _____ *Smer:* _____ *Sala:* _____

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

<i>Zadatak 1</i>	_____ /10	<i>Zadatak 4</i>	_____ /12
<i>Zadatak 2</i>	_____ /10	<i>Zadatak 5</i>	_____ /10
<i>Zadatak 3</i>	_____ /8	<i>Zadatak 6</i>	_____ /10

Ispit: _____ /60

Ukupno: _____ /100

Projekat: _____ /40

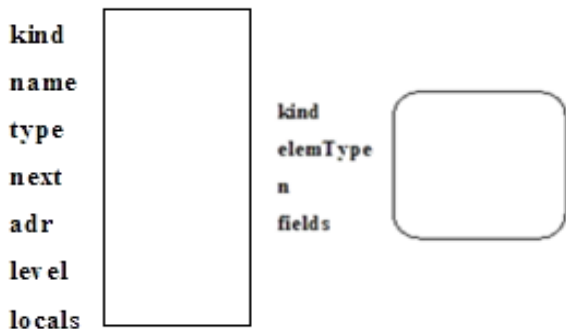
Ocena: _____

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!

Podsetnik za neke instrukcije Mikrojava bajtkoda

getstatic	s	... → ..., global[s]
putstatic	s	..., val → ...
new	s	... → ..., adr
getfield	s	..., adr → ..., adr.fields[s]
putfield	s	..., adr, val → ...
const	w	... → ..., w
load	b	... → ..., local[b]
store	b	..., val → ...
new	s	...
		..., adr
newarray	b	..., n
		..., adr
aload		..., adr, index
		..., val
astore		..., adr, index, val
		...
baload		..., adr, index
		..., val
bastore		..., adr, index, val
		...
enter	b1, b2	
dup		..., val
		..., val, val
dup2		..., v1, v2
		..., v1, v2, v1, v2
dup_x1	..,val2, val1	...,val1, val2, val1
dup_x2	val1, val2, val3	...,val3, val1, val2, val3
		...

Podsetnik strukture čvorova tabele simbola.



1) (10 poena)

Dat je deo implementacije klase `Node` u programskom jeziku *Java*, koja predstavlja čvor u sintaksnom stablu izgenerisanom za konstrukciju DKA korišćenjem metoda pozicije, gde su objašnjenja polja i metoda data u komentarima iznad njih (sva polja su inicijalizovana konstruktorom):

```
public class Node {
    private enum Type {
        ASTERISK,    // *
        PLUS,        // +
        PERIOD,      // .
        OR,          // |
        SYMBOL,      // |
        EPSILON      // ε
    }

    // left subtree of position node, null if leaf (symbol) node
    private Node left;

    // right subtree of position node, null if unary operation or leaf (symbol) node
    private Node right;

    // type of position node
    private Type type;

    // position of symbol in regex, only valid for leaf (symbol) nodes, otherwise set
    // to -1
    private int symbolPos;

    // check whether the subsequence with the given node as root of the subtree can be
    // empty
    public boolean isNullable();

    // find set of leaf (symbol) node positions which could be at the beginning of
    // subsequence with the given node as root of the subtree
    public Set<Integer> firstPos();

    // find set of leaf (symbol) node positions which could be at the end of
    // subsequence with the given node as root of the subtree
    public Set<Integer> lastPos();

    // returns a union of the two parameters
    public static Set<Integer> union(Set<Integer> first, Set<Integer> second);
}
```

Potrebno je napisati implementaciju metode `isNullable()`.

Rešenje:

2) (10 poena)

Zadata je gramatika

1. $\langle S \rangle \rightarrow \langle S \rangle \langle A \rangle$
2. $\langle S \rangle \rightarrow \langle A \rangle a \langle A \rangle b$
3. $\langle A \rangle \rightarrow \langle S \rangle a$
4. $\langle A \rangle \rightarrow c$

- a) Konstruisati karakteristični automat LALR(1) parsera uključujući predikcione skupove.
- b) Konstruisati kontrolnu tabelu automata pod a). Da li ima konflikata?

Rešenje:

3) (8 poena)

a) Data je gramatika: Startni neterminal je $\langle S \rangle$.

1. $\langle S \rangle \rightarrow a \langle S \rangle b c$

2. $\langle S \rangle \rightarrow d$

Modifikovati datu gramatiku dodavanjem u nju neterminala $\langle Z \rangle$ koji je živ, ali nije dostižan i neterminala $\langle M \rangle$ koji je dostižan, ali je mrtav, tako da se ne promeni jezik koji gramatika opisuje. Modifikovana gramatika ne bi trebalo da ima više od 6 smena (rešenje može da se nađe i sa manje smena) i startni neterminal ostaje $\langle S \rangle$.

b) Sledeće smene su deo neke atributivne gramatike. U svakoj od njih, p , q i r su nasleđeni atributi, a s i t sintetizovani (za u i v nije potrebno reći šta su). Gramatika treba da je L-atributivna. Dodati za svaku smenu sva potrebna atributivna pravila u formatu $x \leftarrow f_1(y,z,w)$ $y \leftarrow f_2(\dots)$ itd. U parametrima funkcija f_1 , f_2 , itd treba da se pojavljuju svi mogući atributi od kojih može da zavisi određišni atribut pravila tako da se ne narušava L-atributivnost gramatike.

1. $\langle S \rangle_{s,q} \rightarrow a_u \langle S \rangle_{t,p} \langle X \rangle_r$

2. $\langle X \rangle_r \rightarrow \{d_p\} b_v \langle B \rangle_{t,r}$

3. $\langle B \rangle_{s,q} \rightarrow c_u \langle X \rangle_r \langle B \rangle_{t,p}$

Rešenje:

4) (12 poena)

- a) Sledeći deo mikrojava programa prevesti u mikrojava bajtkod. Pretpostavka je da su sve promenljive globalne. Koristiti simbolička imena promenljivih za reprezentaciju adresa promenljivih.

$$c = (-b) + (c + a)$$

$$b = c - 2$$

- b) Bajtkod dobijen u tački a) prevesti u međukod zasnovan na troadresnim naredbama (koristiti format troadresnih naredbi definisan na predavanjima i vežbama). Algoritam za to "simulira" stek izraza nizom privremenih promenljivih t_1, t_2, t_3 , itd. Ažurira se brojač TC koji u svakom trenutku daje informaciju o indeksu privremene promenljive koja predstavlja vrh steka izraza u datom trenutku prevođenja bajtkoda. Inicijalna vrednost TC je 0 jer je stek izraza na početku prazan. Ako se prevodi bajtkod naredba koja radi neki push na stek izraza, tada se TC uveća za jedan i potom kao odredišni operand odgovarajuće troadresne naredbe iskoristi privremena promenljiva sa indeksom TC. Ako se prevodi bajtkod naredba koja radi neki pop sa steka izraza, kao izvorišni operand troadresne naredbe koristi se privremena promenljiva sa indeksom TC, a zatim TC umanjuje za jedan.
- c) Odrediti informacije o životu i sledećem korićenju za kod dobijen u tački b).
- d) Prevesti rezultat tačke b) u x86 mašinski kod korišćenjem jednostavnog generatora koda sa getreg opisanog na predavanjima i vežbama. Napomena: NEG r/m je x86 naredba za promenu znaka operanda. Na raspolaganju su dva procesorska registra AX i BX.

Rešenje:

5) (10 poena)

Dat je listing programa Avgust2023, napisan na programskom jeziku Mikrojava. Sve metode unutrašnjih klasa su virtuelne. Globalne metode se pozivaju statički. Napisati kompletan Mikrojava bajtkod prevoda funkcije main.

Rešenje:

```
program Avgust2023
const int N = 5;

class A {
    int arr[];
    {
        int meth(int index){
            return arr[index] * 2;
        }
    }
}

class B extends A {
    int factor;
    *T1*
    {
        int meth() {
            return arr[index] * factor;
        }
    }
}
*T2*
B obj;

void main() int i, sum;{
    sum = 0;
    obj = new B;
    obj.factor = 5;
    obj.arr = new int[N];
    while(i < N && sum < N * 10) {
        obj.arr[i] = i;
        sum = sum + obj.meth();
        i++;
    }
}
```

6) (10 poena)

Prikazati izgled tabele simbola u trenucima T1 i T2 na osnovu izvornog koda iz zadatka 5) prema formatu čvorova tabele simbola iz priloga. Universe opseg ne treba crtati.

Rešenje:

Dodatni prostor za rad: