Programski Prevodioci 1 (IR4PP1, SI4PP1) Septembar 2009.

Ispit traje 3 sata, upotreba literature nije dozvoljena.

1. [8] Navesti i objasniti uslove koje atributivno translaciona gramatika mora da zadovolji za parsiranje od dna ka vrhu.
2. [8] Objasniti povratnu transformaciju iz SSA u obični troadresni međukod.
3. [6] Konstruisati minimalni deterministički konačni automat koji prihvata sve i samo one sekvence oblika $\mathbf{s_1 s_2}$ gde je $\mathbf{s_1}$ sekvenca koju prihvata automat sa slike (a), a $\mathbf{s_2}$ sekvenca koju prihvata automat sa slike (b). Jasno naznačiti korake dolaženja do rešenja.

	A	*	
→ A	B		1
B	C	B	0
C		A	1

(a)

	*	&	
→ A1	B1	C1	0
B1	C1		1
C1		A1	0

(b)

4. [7] a) [3] Eliminisati mrtve i nedostižne neterminale iz date gramatike sa startnim simbolom <A>.
   b) [2] Iz gramatike dobijene pod a) eliminisati levu rekurziju.
   c) [2] Izvršiti levu faktorizaciju u gramatici dobijenoj pod b).

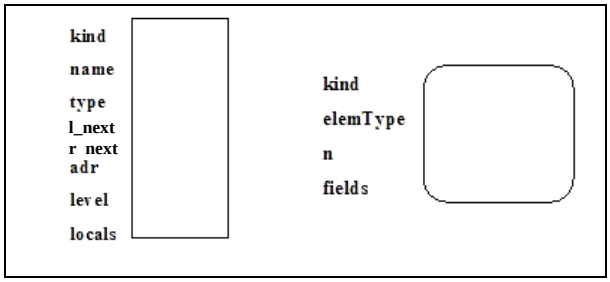
```
1. <A> → <A> c <B>          7. <C> → <C> c <E>
2. <A> → <C> c b            8. <D> → <D> a <B>
3. <A> → c <D>              9. <D> → <B> b <B>
4. <A> → <D>                10. <D> → <B>
5. <B> → b <B>              11. <E> → a b
6. <B> → d
```

5. [8] Data je S-atributivna gramatika poljskog prevodjenja, koja opisuje aritmetičke izraze. Atribut tokena I je pokazivač na tabelu simbola, na simbol koji odgovara tokenu I. Gramatika vrši prevodjenje u niz sastavljen od atoma $ADD_{p\,q\,r}$ i $MULT_{p\,q\,r}$ gde su p, q i r respektivno pokazivači na levi operand, desni operand i rezultat operacije u tabeli simbola. Novi ulaz u tabelu simbola generišemo funkcijom NEWT.
   a) [4] Projektovati bottom-up LR(0) procesor koji će obaviti ovo prevodjenje.
   b) [4] Prikazati rad procesora, po koracima, prilikom prepoznavanja sledeće sekvence na ulazu: *(a + (b + c))*. U svakom koraku prikazati izgled steka, preostali ulaz, akciju koja se vrši i kod koji se generiše.

```
1. <E>p → ( <E>q + <E>r ) {ADD}s1,s2,s3
     s1←q,  s2←r,  (p,s3)←NEWT
2. <E>p → Ir
     p←r
```

6. [6] Za dati MikroJava program, nacrtati izgled tabele simbola nakon parsiranja kompletnog koda. 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 ABC
  int c
{
  int f (int a, int b)
  {
    a++;
    print(a, 5);
    return b;
  }
  void main() int a;
  {
    a = 2; c = 4;
    while(a < 0) { f(c, a); }
  }
}
```



7. [9] Dat je iskaz `a := ar[x+5]*b + -b/c*H - m.n`
   a) [3] Nacrtati sintaksno stablo.
   b) [3] Prevesti iskaz u troadresni kod. Neka za niz **ar** važi da mu je donji indeks 1, početna adresa neka konstanta **base1**, a veličina jednog elementa 4 adresibilne jedinice. Za strukturu **m** važi da je njena početna adresa konstanta **base2**, veličina pojedinačnog polja 4 adresibilne jedinice, pozicije polja se numerišu od 0, a polje **n** se nalazi na 2. poziciji.
   c) [3] Prikazati implementaciju troadresnog koda iz tačke b) putem četvorki.
8. [8] Prikazati sve promene hipa i Procesorskog Steka (ProcStack), za dati MikroJava program.

```
int sumiraj(int i)
  int a[];
{
  a = new  int[3];
  a[0] = 12;
  i = 0;
  return i;
}
```

load b	... → ..., val
putstatic s	..., val → ...
getstatic s	... → ..., val
putfield s	..., adr, val → ...
const w	... → ..., val
add	..., val1, val2 → ..., val1+val2
new s	... → ..., adr
aload	..., adr, index → ..., val