

Pismeni deo ispita iz Programskih prevodilaca

1. Naredba za deklaraciju promenljivih u programu je definisana sledećom gramatikom:

$$\begin{aligned} Dec &\rightarrow \mathbf{TYPE} \text{ } VarL \ ; \\ VarL &\rightarrow Var \mid VarL \ , \ Var \\ Var &\rightarrow \mathbf{ID} \mid \mathbf{ID} = \mathbf{CONST} \end{aligned}$$

- Transformisati datu gramatiku u LL(1) gramatiku i dokazati da tako transformisana gramatika jeste LL(1) gramatika.
- Kreirati LR sintaksnu tabelu za zadatu gramatiku.

2. Kreirati cup specifikaciju za generisanje sintaksnog i semantičkog analizatora naredbe za deklarisanje promenljivih. Za leksičku analizu koristiti analizator generisan pomoću flex-a što podrazumeva da treba kreirati i odgovarajuću flex specifikaciju. Značenja tokena su sledeća:

TYPE – ime tipa: **char**, **int**, **double**

ID – identifikator (niz slova cifara i ‘_’ u kojem prvi znak ne može da bude cifra)

CONST – konstata tipa char, int ili double (formata odgovarajućih konstanti u C-u)

Semantička pravila ovog jezika su:

- Promenljiva zadatog imena može biti deklarisan samo jednom.
- Ukoliko se u deklaraciji postavlja i inicijalna vrednost promenljive, mora da se slaže tip promenljive i konstante koja joj se dodeljuje.

3. Definirati klasu za predstavljanje select naredbe u apstraktnom sintaksnom stablu. Select naredba je definisana sledećom gramatikom:

$$\begin{aligned} SelectStatement &\rightarrow \mathbf{select} \text{ } AlternativeList \ \mathbf{end} \\ AlternativeList &\rightarrow AlternativeList \ \mathbf{or} \ Alternative \mid Alternative \\ Alternative &\rightarrow \mathbf{when} \text{ } Expression \Rightarrow Statement \end{aligned}$$

Značenje naredbe: Izvršiće se prva alternativa čiji je uslov zadovoljen (tj. čiji izraz ima vrednost različitu od nule). Ukoliko nijedan od navedenih uslova nije zadovoljen, izvršenje programa se nastavlja od prve naredbe navedene nakon ove strukture.

Definisati zapis select naredbe u međukodu niskog nivoa, i u definisanoj klasi za predstavljanje ove naredbe u apstraktnom sintaksnom stablu implementirati funkciju *translate* za generisanje takvog međukoda.

4. Data je funkcija za nalaženje sume elementa celobrojnog niza.

```
int sum(int n, int* a)
{
    int s, i;
    if ( n==0 )
        s=0;
    else
        s=*a + sum(n-1, a+1);
    return s;
}
```

Definisati sadržaj aktivacionog sloga funkcije *sum* i 8086 asemblerski kod za datu funkciju. Pretpostaviti da se rezultat funkcije upisuje u registar CX.

I kolokvijum iz Programskih prevodilaca

1. Naredba za deklaraciju promenljivih u programu je definisana sledećom gramatikom:

$$\begin{aligned} Dec &\rightarrow \text{TYPE } VarL ; \\ VarL &\rightarrow Var \mid VarL , Var \\ Var &\rightarrow ID \mid ID = CONST \end{aligned}$$

- Transformisati datu gramatiku u LL(1) gramatiku.
- Dokazati da tako transformisana gramatika jeste LL(1) gramatika.
- Kreirati LL(1) sintaksnu tabelu date gramatike.
- Korišćenjem kreirane sintaksne tabele proveriti da li je sledeća deklaracija ispravno zapisana:

TYPE ID = CONST, ID ;

2. Kreirati LR sintaksnu tabelu za gramatiku datu u prvom zadatku i korišćenjem te tabele proveriti da li je deklaracija navedena u prvom zadatku korektno zapisana.

II kolokvijum iz Programskih prevodilaca

3. Definirati klasu za predstavljanje select naredbe u apstraktnom sintaksnom stablu. Select naredba je definisana sledećom gramatikom:

$$\begin{aligned} SelectStatement &\rightarrow \text{select } AlternativeList \text{ end} \\ AlternativeList &\rightarrow AlternativeList \text{ or } Alternative \mid Alternative \\ Alternative &\rightarrow \text{when } Expression \Rightarrow Statement \end{aligned}$$

Značenje naredbe: Izvršiće se prva alternativa čiji je uslov zadovoljen (tj. čiji izraz ima vrednost različitu od nule). Ukoliko nijedan od navedenih uslova nije zadovoljen, izvršenje programa se nastavlja od prve naredbe navedene nakon ove strukture.

Definisati zapis select naredbe u međukodu niskog nivoa, i u definisanoj klasi za predstavljanje ove naredbe u apstraktnom sintaksnom stablu implementirati funkciju *translate* za generisanje takvog međukoda.

4. Data je funkcija za nalaženje sume elementa celobrojnog niza.

```
int sum(int n, int* a)
{
    int s, i;
    if ( n==0 )
        s=0;
    else
        s=*a + sum(n-1, a+1);
    return s;
}
```

Definisati sadržaj aktivacionog sloga funkcije *sum* i 8086 asemblerski kod za datu funkciju. Pretpostaviti da se rezultat funkcije upisuje u registar CX.