

Pismeni deo ispita iz Programskih prevodilaca

1. For-petlja u jednom programskom jeziku definisana je sledećom gramatikom:

ForStatement \rightarrow **for** ID **in** *Expressions* **do** *Statement* **end**

Expressions \rightarrow (*ExprList*)

ExprList \rightarrow *ExprList* , *Expr* | *Expr*

Expr \rightarrow ID | CONST

- Transformisati datu gramatiku u LL(1) gramatiku i dokazati da tako transformisana gramatika jeste LL(1) gramatika.
 - Kreirati LR sintaksnu tabelu date gramatike.
2. Kreirati *cup* specifikaciju za generisanje sintaksnog (i semantičkog) analizatora for-petlje koja je opisana gramatikom iz prethodnog zadatka. Za generisanje potrebnog leksičkog analizatora kreirati odgovarajuću *flex* specifikaciju. Terminalni simbol ID predstavlja niz cifara i slova u kojem je prvi znak obavezno slovo. CONST je celobrojna konstanta zapisana u dekadnom brojnom sistemu, ili string (niz karktera između dvostrukih navodnika). Semantičko pravilo o kojem treba voditi računa je da tip promenljive ID mora da se slaže sa tipovima svih izraza u listi.
3. Definisati klasu za predstavljanje for petlje definisane gramatikom datom u prvom zadatku. Definisati međukod niskog nivoa koji odgovara takvoj for petlji i u klasi koja for petlju predstavlja u apstraktnom sintaksnom stablu implementirati funkciju za generisanje takvog međukoda.
4. Data je funkcija za izračunavanje maksimalnog elementa niza:

```
int max( int *p, int n )
{
    int q;
    if( n == 1 ) return *p;
    q = max( p+1,n-1);
    if( *p > q ) return *p;
    return q;
}
```

- Definisati sadržaj aktivacionog sloga funkcije *max*.
- Definisati 8086 asemblerski kod za datu funkciju. Pretpostaviti da se rezultat funkcije prenosi kroz listu parametara.