

Programski prevodioci

1. (10 poena) Tip liste u jednom programskom jeziku definisan je sledećom gramatikom:

ListDefinition → **list** (*Elements*)

Elements → *Elements* , *Element* | *Element*

Element → **NAME** = *Value*

Value → **NAME** | **CONST** | *ListDefinition*

- a) Transformisati datu gramatiku u LL(1) gramatiku i dokazati da tako transformisana gramatika jeste LL(1) gramatika.
b) Kreirati LR sintaksnu tabelu date gramatike.

2. (16 poena) Kreirati *cup* specifikaciju za generisanje sintaksnog (i semantičkog) analizatora deklaracije tipa liste koji je opisan gramatikom iz prethodnog zadatka. Za generisanje potrebnog leksičkog analizatora kreirati odgovarajuću *flex* specifikaciju. Terminalni simbol **NAME** predstavlja niz brojki i slova u kojem je prvi znak obavezno slovo, a **CONST** može biti celobrojna, realna, logička ili string konstanta (čiji format odgovara formatu konstanti u programskom jeziku C).
Semantičko pravilo o kojem treba voditi računa: 2 elementa liste ne mogu imati isto ime.

3. (6 poena) „Switch izraz“ u jednom programskom jeziku je definisan na sledeći način:

SwitchExpression → **switch** (*Expression* , *ExpressionList*)

Definisati klasu za predstavljanje „switch izraza“ u apstraktnom sintaksnom stablu. *Exp se izračuna => INT*
Definisati međukod niskog nivoa za izračunavanje vrednosti „switch izraza“ i u klasi koja ovaj izraz predstavlja u apstraktnom sintaksnom stablu implementirati funkciju za generisanje takvog međukoda. *INT - ne mora da postoji u EXE list*

Značenje izraza je sledeće: izračunava se vrednost prvog izraza u zagradi i to predstavlja redni broj izraza iz liste čija će vrednost biti vraćena kao rezultat.

4. (8 poena) Definisati aktivacioni slog i odgovarajući 8086 kod za funkciju sum. Pretpostaviti da se rezultat funkcije prenosi kroz listu parametara.

```
struct node
{
    int info;
    struct node* next;
};
int sum( struct node* head )
{
    int s=0;
    if (head != 0)
        s = head->info + sum(head->next);
    return s;
}
```