

### Programski prevodioci

1. (10 poena) Jedan HTML element definisan je sledećom gramatikom:

$Element \rightarrow \text{STARTTAG } Content \text{ ENDTAG}$

$Content \rightarrow \text{TEXT} \mid ElementList$

$ElementList \rightarrow ElementList \text{ Element} \mid Element$

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

2. (16 poena) Kreirati *cup* specifikaciju za generisanje sintaksnog (i semantičkog) analizatora definicije tabele u HTML-u. Za generisanje potrebnog leksičkog analizatora kreirati odgovarajuću *flex* specifikaciju. Terminalni simboli su definisani na sledeći način:

- STARTTAG** – ime elementa napisano kao niz malih slova (jednostavnosti radi smatramo da je jezik *case sensitive*) između simbola `< i >`. Od interesa su samo elementi: `table`, `tr`, `td`, `th`.
- ENDTAG** – ime elementa napisano između simbola `</ i >`.
- TEXT** – bilo koji niz karaktera koji ne sadrži simbole `< i >`.

Semantička pravila jezika su:

- Elementi se mogu samo ugnježdavati, ne mogu se preklapati (imena početnog i završnog taga prilikom primene jedne smene moraju biti ista),
- Sadržaj elementa **table** može biti samo niz elemenata **tr**.
- Element **tr** može sadržati isključivo elemente **td** i **th**.
- Sadržaj elemenata **td** i **th** može biti isključivo **TEXT**.
- Elementi **th** se mogu naći samo u prvoj vrsti ili prvoj koloni tabele.

3. (6 poena) „While petlja“ u nekom programskom jeziku je definisana na sledeći način:

$WhileLoop \rightarrow \text{while Expression: StatementList else Statement}$

Definisati klasu za predstavljanje „while petlje“ u apstraktnom sintaksnom stablu. Definisati međukod niskog nivoa za izvršavanje ovakve petlje i implementirati metodu za generisanje takvog koda. Značenje ovakve strukture je: niz naredbi iz tela petlje se izvršava dok je uslov zadovoljen. Naredba iz else grane se izvršava kada uslov nije zadovoljen (svejedno da li uslov nije zadovoljen na početku i petlja se uopšte ne izvrši ili posle nekog broja izvršenja petlje uslov prestane da važi).

4. (8 poena) Definisati 8086 kod i izgled aktivacionog sloga za funkciju *apply\_strategy*. Za predstavljanje podataka tipa `int` kao i memorijskih adresa koriste se 2 bajta. Pretpostaviti da se rezultat funkcije *apply\_strategy* smešta na stek, a rezultat funkcije *exec\_query* u registar CX.

```
struct adaptation_rule{
    int condition, action;
    struct adaptation_rule* next;
};

int apply_strategy(int env_change, struct adaptation_rule* ar){
    int strategy=0, condition;
    condition=exec_query(ar->condition);
    if (condition == env_change)
        strategy = ar->action;
    if(ar->next != 0)
        strategy += apply_strategy(env_change, ar->next);
    return strategy;
}
```