

Prevodioci i interpretatori - Mart 2011.
praktični deo

1. Konstruisati MDKA za prepoznavanje jezika nad azbukom $\Sigma = \{a, b\}$ čija svaka reč ne sme da sadrži faktor *abab*, i zatim napisati C program koji proverava da li uneta niska pripada ovom jeziku.
2. Opisati deklaracije klasa u C++-u LL(1) gramatikom, a zatim napisati perl skript koji metodom rekurzivnog spusta proverava ispravnost deklaracije.

```
class Vektor {
public:
    double getX();
    void setX(double x);
    double norm();
private:
    double x;
    double y;
};
```

```
class Function {
public:
    double x;
private:
    void deinit();
};
```

3. Napisati interpretator za jezik koji dopušta definisanje matematičkih funkcija. izračunavanje njihovih vrednosti i pronalaženje njihovih izvoda. Funkcije se definišu aritmetičkim operacijama sabiranja i množenja i kompozicijom elementarnih funkcija x , $\sin(x)$, $\cos(x)$ i konstantnih funkcija. Npr. $\sin(\cos(x) + x) * \cos(x) + 3.14$. Komande jezika se navode u posebnim linijama.

- (a) Realizovati računanje vrednosti aritmetičkog izraza koji u sebi sadrži sabiranje, oduzimanje, množenje, deljenje, kao i funkcije sinus i kosinus nad realnim brojevima. Vrednost izraza se izračunava komandom *value*.

value "23.01 - 11 * 2"	1.01
value "sin(3.14) - (11 + cos(0))"	-11.9984

- (b) Omogućiti definisanje funkcija jedne promenljive. Imena funkcija moraju biti niz malih slova abecede. Funkcije se zadaju lambda računom. (ključna reč lambda prethodi nezavisnoj promenljivoj koja figuriše u izrazu funkcije) Štampanje funkcija se vrši prostim navođenjem njenog imena.

function f = lambda x. 1 + x * x	
f	(1)+((x)*(x))
function g = lambda y. sin(cos(y))+3.12	
g	(sin(cos(x)))+(3.12)

- (c) Implementirati računanje izvoda funkcije. Izvod se označava apostrofom.

function h = lambda x. (1+x*x)'	
h	(0)+(((1)*(x))+((x)*(1)))
f'	(0)+(((1)*(x))+((x)*(1)))

- (d) Računanje vrednosti funkcije u nekoj tački realizuje se operatorom [].

h[4]	8
------	---

Srećno!