

# Tema 1: Analizor lexical

Tema de laborator in lucru: 3 saptamani

## Enunt general:

Scrierea unui ANALIZOR LEXICAL pentru un minilimbaj de programare (MLP),  
**ales ca subset al unui limbaj existent.**

Obs.: Va rugam sa consultati standardul limbajului ales.

De exemplu, pentru C++, puteti folosi:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/n4659.pdf>

iar pentru Pascal: <http://pascal-central.com/iso7185.html>

## 1. Specificarea minilimbajului de programare (MLP).

Limbajul trebuie sa contina cel putin anumite instructiuni si tipuri de date:

- 2 tipuri de date simple si un tip de date definit de utilizator
- instructiuni:
  - o instructiune de atribuire
  - o instructiune de intrare/iesire
  - o instructiune de selectie (conditionala)
  - o instructiune de ciclare

Pe langa acestea, vor exista unele restrictii suplimentare referitoare la identificatori si constante (vezi sectiunea 3.1).

Se cere ca specificarea sa respecte structurile sintactice ale limbajului ales, intr-o forma (foarte) simplificata. De asemenea, ea va fi suficient de generala astfel incat sa descrie constructiile (limbajului) folosite pentru scrierea programelor de la pct.1

## 2. se cer textele sursa a 3 mini-programe

care respecta specificatiile MLP date si care rezolva urmatoarele probleme:

- calculeaza perimetrul si aria cercului de o raza data
- determina cmmdc a 2 nr naturale
- calculeaza suma a n numere citite de la tastatura

Se cere ca mini-programele sa fie compilate si executate in limbajul original ales.

## 3. Se cer textele sursa a doua programe care contin erori conform MLP-ului definit:

- Unul dintre programe contine doua erori care sunt in acelasi timp erori in limbajul original (pentru care MLP defineste un subset)
- Al doilea program contine doua erori conform MLP, dar care nu sunt erori in limbajul original. Se cere ca acesta sa fie compilat si executat in limbajul original ales.

## 4. Implementarea analizorului lexical

Analizorul lexical accepta la intrare un fisier text reprezentand un

program sursa si intocmeste ca date de iesire tabelele:

FIP - forma interna a programului sursa si

TS - tabelui de simboluri.

In plus, programul va trebui sa semnaleze erorile lexicale si locul in care apar.

Niste restrictii suplimentare vor fi impuse analizoarelor lexicale.

Acestea **se vor diferentia dupa** urmatoarele criterii:

1. tabela de simboluri:

a. unica pentru identificatori si constante

b. separat pentru identificatori si constante

2. organizarea tabelelor de simboluri:

a. tabel ordonat lexicografic

b. tabel arbore binar de cautare (ordine lexicografica)

c. tabel de dispersie (hash)

Se cer, de asemenea:

- implementarea structurilor de date cerute pentru tabela de simboluri;  
- **organizarea FIP si TS trebuie aleasa astfel incat accesul la informatiile pentru un atom lexical in TS sa se faca in Theta(1) pe baza informatiilor din FIP.**

- salvarea FIP si TS in fisiere. Pozitia in TS (din fisierul corespunzator FIP) va fi numarul liniei din fisierul TS in care este stocata informatia referitoare la acel atom lexical.

### **Descompunerea temei in parti (laboratoare), deadline-uri , alte activitati si punctaje:**

<b>Teme partiale</b>		<b>Termene de realizare</b>	<b>Punctaje</b>
Partea intai	MLP (cerintele 1, 2 si 3)	Sapt.2	2 puncte
	Activitate in timpul laboratorului	Sapt.2	1 punct
Partea a doua	Program ce identifica atomii lexicali (ce vor fi trimisi spre analiza sintactica) si prezentarea acestora (prin afisare pe ecran scriere in fisier, etc.)	Sapt.3	3 puncte
Partea a treia	Analizorul lexical complet, conform cerintei 4.	Sapt.4	3 puncte
		Din oficiu pt. program predat	1 punct

Total: 10 puncte