**Tema 3: Utilizare LEX/FLEX**

Documentatie

# Cerinta

Folositi LEX/FLEX pentru a obtine un analizor lexical.

* Scrieti o specificatie LEX/FLEX continand expresiile regulare corespunzatoare atomilor lexicali ai minilimbajului definit pt. Laboratorul nr. 1.
* Testati analizorul pentru aceeasi intrare ca in lab 1.
* Programul trebuie sa respecte aceeasi specificatie (date/rezultate) cu analizorul lexical din laboratoarele precedente.
* Puteti utiliza oricare dintre versiuni (LEX sau FLEX) pentru generarea analizorului lexical.

# Detalii de implementare

Pentru rezolvarea cerintei s-a folosit FLEX, in sistemul de operare Linux. S-au definit fisierele analizator.lex, Codare.h, BST.h, Analiza.h.

Fosierul analizator.lex este entry point-ul programului si contine regulile LEX. La fel ca la laboratoarele anterioare, exista doua tipuri de constante (string si intreaga):

CONSTANTA\_STRING \”.\*\”

Si

CONSTANTA\_INTREAGA {CIFRA}|{CIFRA\_NENULA}{CIFRA}\*

CIFRA [0-9]

CIFRA\_NENULA [1-9]

Regula corespunzatoare identificatorului este

IDENTIFICATOR [a-zA-Z\_][a-zA-Z0-9\_]{0,7}

Mai intai se verifica regula corespunzatoare constantei string, urmata de regulile pentru operatori, sepacatori, cuvinte cheie, identificatori, constanta intreaga. Aceste reguli sunt dupa urmate de alte doua:

[\n]+ => regula ce elimina new line-ul

. => semnaleaza o eroare lexicala, in cazul in care niciuna din reguluile anterioare nu s-a portivit

Fisierul Codare.h contine definitia unui atom. Acesta are un intreg ce reprezinta codul sau, si un dataCod care este exact denumirea atomului.

Fisierul BST.h contine definitia arborelui binar de cautare.

Fiserul Analiza.h contine logica din spatele analizei lexicale. In aceasta clasa se declara cei trei arbori ce contin codurile corespunzatoare constantelor, identificatorilor si tutouror atomilor, in general. Tot aceasta clasa ajutatoare efectueaza scrierea tabelelor de simboluri in fisier, codificarea fisierului ce contine codurile, crearea arborilor de cautare.

# Utilizare

Fisierul analizator.lex se compileaza cu flex (*flex analizator.lex*), dupa care se compileaza rezultatul cu g++ (*g++ yy.lex.c*). La linia de comanda se da fisierul out rezultat impreuna cu fisierul de analizat.

Analizatorul incearca sa portiveasca fisierul de input cu regulile existente, daca se ajunge la regula „.”, atunci analizatorul a depistat o eroare lexicala. In cazul in care una din regulile corespunzaotare identificatorilor/cuvintelor cheie/constantelor/operatorilor/separatorilor se potriveste cu inputul, atunci atomul se scrie in FIP (si daca este constanta/identificator si in tabela de simboluri aferenta tipului).

La final, arborii corespunzatori tabelelor de simbolui pentru identificatori si constante se scriu in fisiere.