

README
TEMA 1. LFA.
PARSER SIRURI DE CARACTERE
Calmis Alina , 335CB

Pentru implementarea parserului am utilizat la maxim informatia oferita in primul curs de LFA.

Funcțiile implementate sunt :

- ➔ boyer_moore
- ➔ findLetter
- ➔ print_delta
- ➔ compute_delta

1. boyer_moore

Este functia care se apeleaza din main, folosind, stringul si pattern-ul citite din fisier, fara ultimul character("\n"). Aici se creeaza delta propriu-zisa. Apoi, printr-o iteratie, se verifica existenta fiecărei litere si indicele ei din matricea delta, salvata in variabila **state** . Functia returneaza vectorul **result** cu indicii aferenti ai pattern-ului gasiti in string.

2. find_letter

In lista **alphabet** este salvat tot alfabetul de la A la Z, printr-o iteratie se gaseste indexul literei cautate, si se returneaza valoarea sa + 1, din motiv ca matricea delta are o anumita structura , exemplificata mai jos.

3. print_delta

Functie aditionala, care prindeaza delta pentru verificare.

4. compute_delta

In prima parte se construiesc matricea delta. Se creeaza alfabetul (**alphabet**), care va fi prima linie a matricei, vectorul patt - salveaza pattern-urile , incepand cu **e** (epsilon), care vor fi si prima coloana a matricei delta. Iar variabila delta[0][0] mereu va fi goala.

	A	B	Z
e					
L					

In cea de-a doua parte a functiei se completeaza matricea delta. Se ia fiecare litera din pattern si se creeaza combinatii cu fiecare litera din alfabet , daca combinatia este egala cu secventa de pattern de lungimea combinatiei , atunci se trece in alta stare, salvand-o in delta. Daca nu atunci se face procesul invers, analizand fiecare element in parte din patternul auxiliar creat(**aux_patt**) , eliminand cate o litera de la inceputul cuvintului, pentru a evita pierderea unei secvente potrivite de la sfarsit.