

TEMA 2

LFA

Calimis Alina 335CB

Pentru implementarea temei am utilizat scheletul din laboratorul 5 cu mici modificari.

Am creat o noua clasa **DFA** cu urmatoarele campuri :

- numberOfStatesD : un intreg ce reprezinta numarul de stari total din DFA
- finalStatesD: un set de numere intregi , in care sunt salvate stările finale
- statesCodificationD : e un vector de seturi de intregi , in care sunt salvate codificarile stărilor noi
- deltaD: dictionar in care sunt salvate noile tranzitii

In structura clasei NFA am creat doua proceduri aditionale :

->**EpsilonClosure()**:

Pentru fiecare stare in parte creeaza o inchidere epsilon. In prima instantă, se adauga insasi starea. Iterativ, pentru fiecare tranzitie din delta si pentru fiecare urmatoare stare se verifica daca este echivalenta cu starea curenta, cat si daca are tranzitie pe epsilon. Se adauga starea in setul intermediar eps, daca corespunde conditiei si verifica daca exista un pas urmator cu tranzitia epsilon a noii stari in delta, daca exista atunci starea curenta devine starea nou gasita.

->**getDFA(closures)**:

Programul incepe de la inchiderea lui 0.

Verifica daca 0 va fi stare finala. Atat timp cat starea curenta (zero_state) este diferita de numarul de stari calculate in proces se realizeaza urmatoarii pasi:

- > Daca zero_state este 0 , atunci continuam cu inchiderea lui 0, in caz contrar, cu noile stari calculate
- > Pentru fiecare litera din alfabet, se verifica sa nu fie epsilon, se creeaza un nou set (new_state) , pentru fiecare stare din inchidere, daca starea si litera sunt instante ale lui delta, atunci in new_state se adauga toate next_step-urile de la tranzitia curenta
- > Se creeaza un set auxiliar, in care se initiaza noua stare, pentru fiecare element adaugand inchiderea .
- > Se salveaza fiecare multime de stari in vectorul vector_of_states, iar codificarea lor in dictionarul final
- > Se verifica daca nu exista deja starea noua in vectorul de stari, daca exista se adauga noua tranzitie cu starea deja existenta in final; in caz contrar, se creeaza o noua codificare a starii (count_states), se adauga in final , cat si in vectorul de stari
- > Se face verificarea corespunzatoare stărilor finale
- > Cand starea curenta devine egala cu numarul de stari noi , se completeaza structura unui DFA si se returneaza DFA-ul creat.