

LFA - Proiect 2

O problema la alegere între:

1. Transformare AFN \rightarrow AFD
2. Transformare AFN-lambda \rightarrow AFD
3. Automatul minimal
4. Verificare generare cuvânt pentru o gramatică regulată
5. Pentru o gramatică regulată, să se genereze toate cuvintele având o lungime maximă dată
- *6. Transformare expresie_regulată \rightarrow automat_finit
- *7. Transformare automat_finit \rightarrow expresie_regulată (algoritm cu eliminarea pe rând a stărilor)

Opțiunea aleasă va fi trecută în documentul aflat pe acest [link](#).

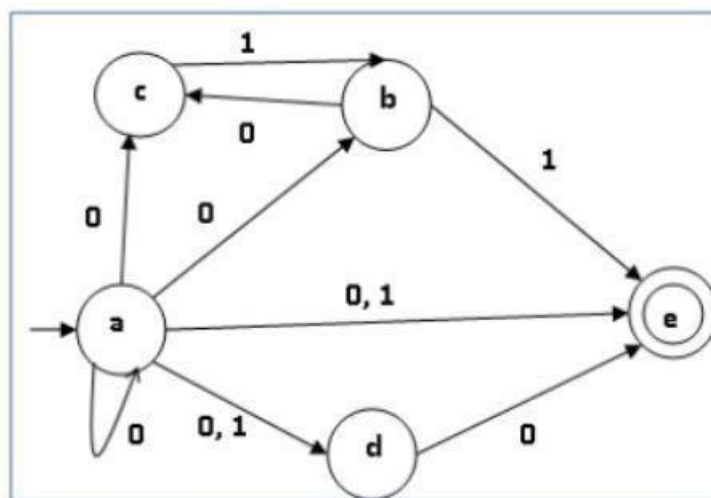
Fiecare semigrupa are o filă separată. Cel mult 4 studenți pot alege aceeași temă în cadrul unei semigrupe.

1. Transformare AFN \rightarrow AFD

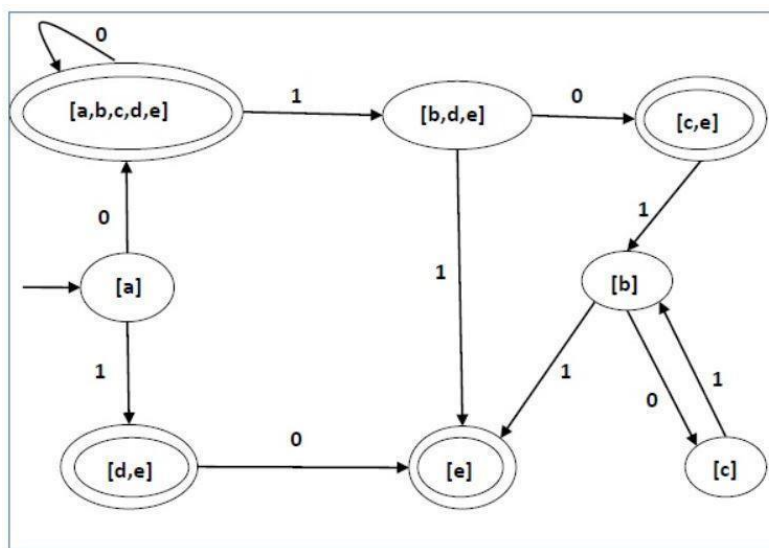
Input: un automat finit nedeterminist oarecare

Output: automatul finit determinist rezultat in urma aplicarii algoritmului de conversie asupra automatului dat ca input.

Exemplu AFN:



AFD rezultat:

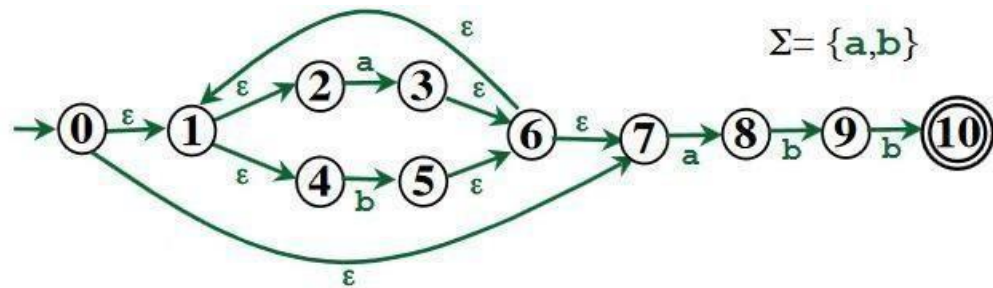


2. Transformare AFN-lambda \rightarrow AFD

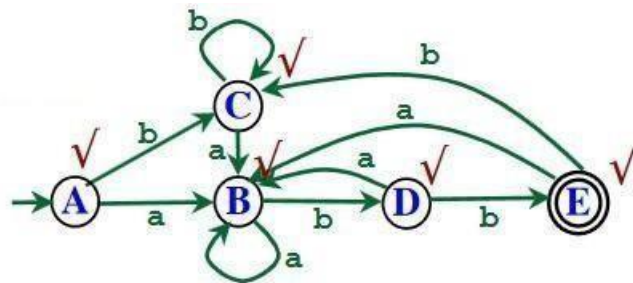
Input: un automat finit nedeterminist cu λ -deplasari oarecare

Output: automatul finit determinist rezultat in urma aplicarii algoritmului de conversie asupra automatului dat ca input

Exemplu λ -AFN:



AFD rezultat:



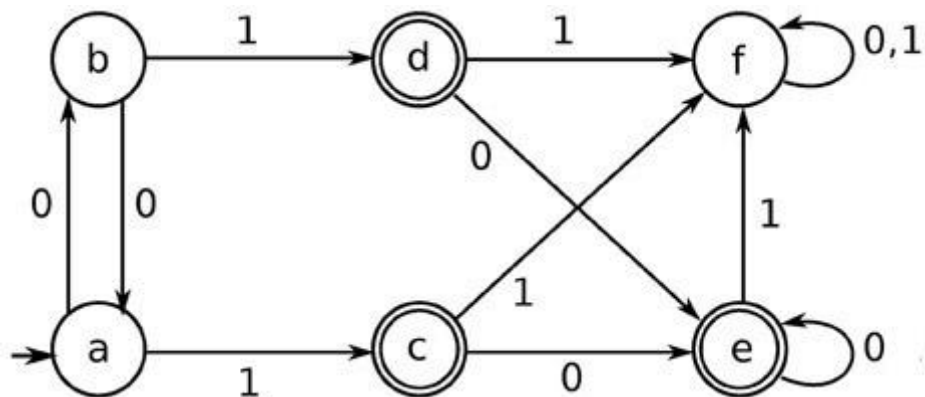
3. Automatul minimal

Input: un automat finit determinist oarecare

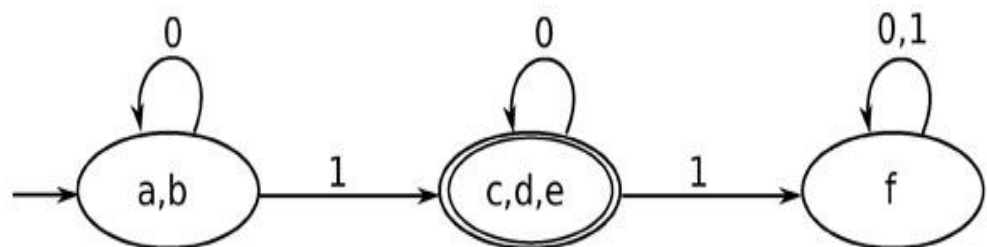
Output: automatul finit determinist minimal rezultat in urma aplicarii algoritmului

Exemplu:

AFD:



AFD minimal:



4. Verificare generare cuvânt pentru o gramatică regulată

Input: O gramatică regulată oarecare și o listă de cuvinte

Output: Pentru fiecare cuvânt dat ca input se precizează dacă acesta aparține limbajului generat de gramatică.

Exemplu:

***G*:**

$N = \{S, A\}$

$\Sigma = \{a, b, c\}$

$S \rightarrow aS$

$S \rightarrow bA$

$A \rightarrow \varepsilon$

$A \rightarrow cA$

$w = ab \in L(G)$

$w = aaaaaaabcccccccc \in L(G)$

$w = b \in L(G)$

$w = bcc \in L(G)$

$w = ac \notin L(G)$

$w = abbc \notin L(G)$

5. Pentru o gramatica regulata, sa se genereze toate cuvintele avand o lungime maxima data

Input: O gramatica regulata oarecare si un numar natural, n .

Output: Lista cuvintelor de dimensiune n recunoscute de gramatica Exemplu:

G :

$N = \{S, A\}$

$\Sigma = \{a, b, c\}$

$S \rightarrow aS$

$S \rightarrow bA$

$A \rightarrow \varepsilon$

$A \rightarrow cA$

$n = 2 \rightarrow ["ab", "bc"]$

$n = 3 \rightarrow ["aab", "abc", "bcc"]$

$n = 4 \rightarrow ["aaab", "aabc", "abcc", "bccc"]$

*6. Transformare expresie_regulata \rightarrow automat_finit

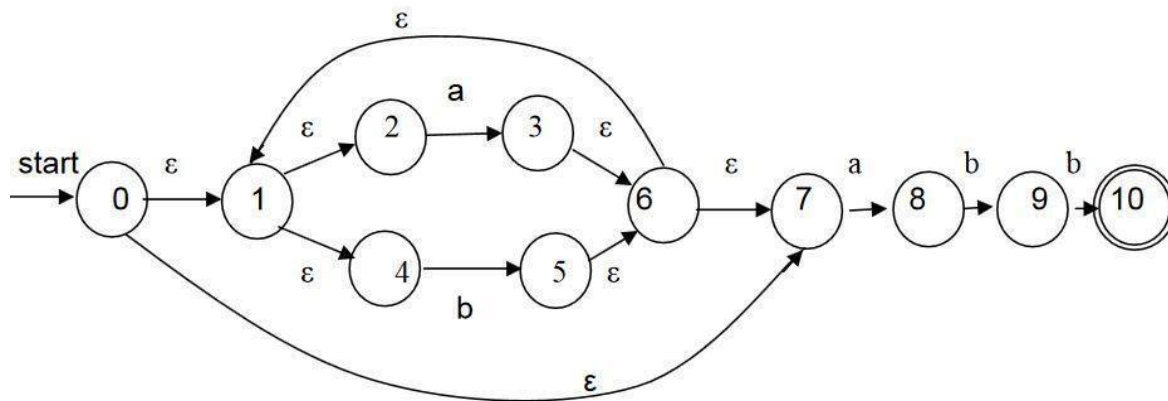
Input: o expresie regulata oarecare

Output: automatul finit echivalent expresiei regulate date la intrare (automatul nu trebuie sa fie neaparat determinist). Intrucat un automat poate avea diverse forme echivalente, este acceptata orice solutie echivalenta cu automatul finit determinist minimal corespunzator expresiei regulate.

Exemplu:

$E = (a|b)^*abb$

Automatul:



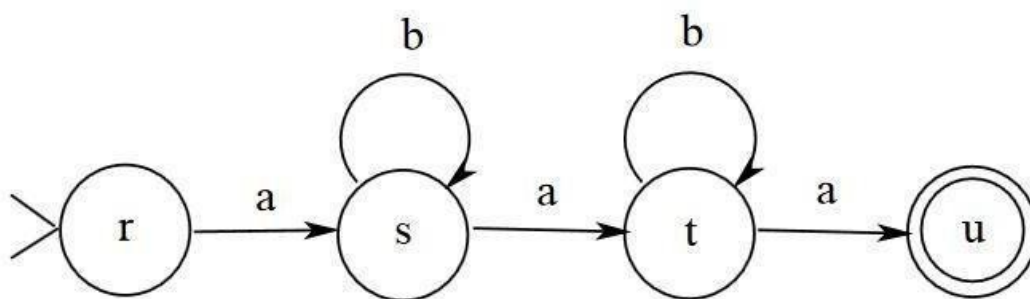
*7. Transformare automat_finit \rightarrow expresie_regulata (algoritm cu eliminarea pe rand a starilor)

Input: Un automat finit oarecare

Output: Expresia regulata echivalenta cu automatul dat ca intrare

Exemplu:

Automatul finit:



Expresia regulata: ab^*ab^*a