

LFA - Proiect 3

O problema la alegere intre:

1. Verificare acceptare cuvant pentru APD
2. Verificare generare cuvant pentru o GIC
3. Pentru o GIC, sa se genereze toate cuvintele avand o lungime maxima data
4. Transformare GIC in Forma Normala Chomsky
5. Algoritmul CYK (verificare generare cuvant de catre gramatica in FNC)

Optiunea aleasa va fi trecuta in documentul aflat pe acest [link](#).

Fiecare semigrupa are o fila separata. Cel mult 4 studenti pot alege aceeași tema în cadrul unei semigrupe. După ce alegeți tema, vă rog să trimiteți și un mail în care să mi-o comunicați. În cazul în care apar conflicte la editarea documentului în momentul alegerii unei teme, voi lua în considerare primele 4 mail-uri trimise pentru alegerea temei respective.

Observatie: Studentii care au ales pentru proiectul 2 problemele 4 sau 5 nu vor putea selecta pentru proiectul 3 problemele 2 sau 3.

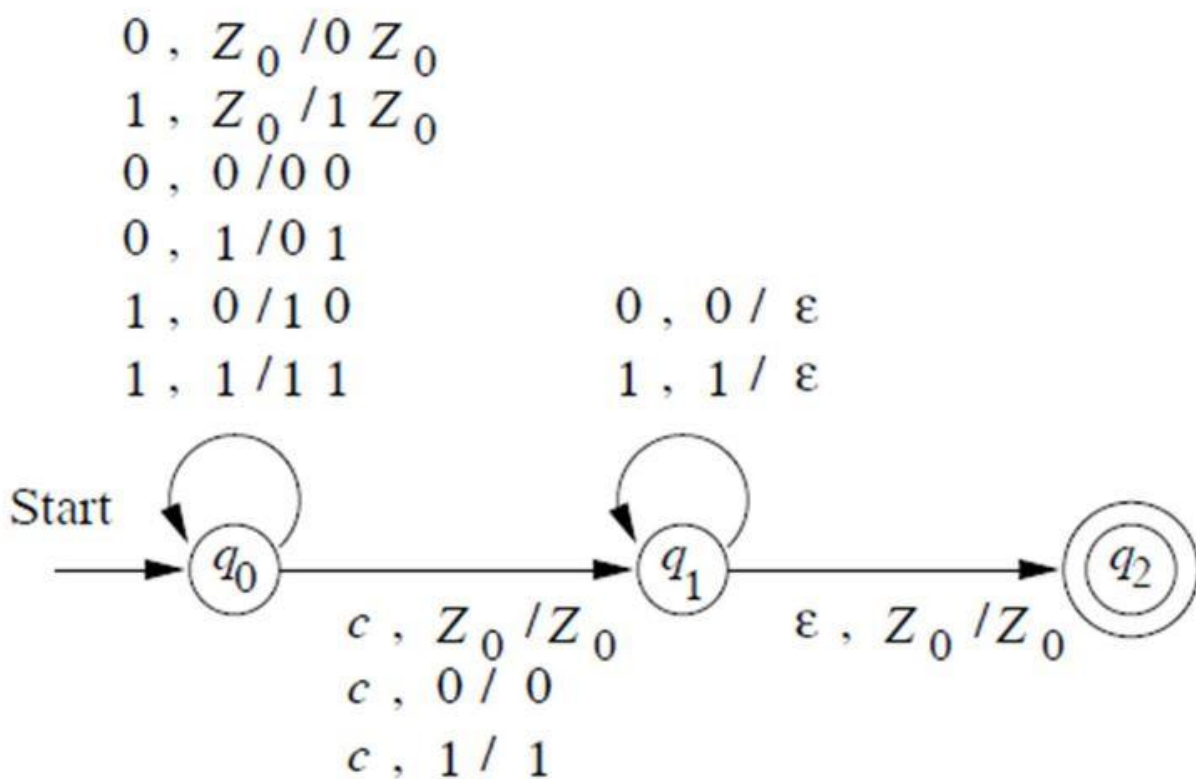
1. Verificare acceptare cuvânt pentru APD

Input: un automat push down oarecare și un cuvânt

Output: boolean (se precizează dacă acel cuvânt a fost sau nu acceptat de automatul dat ca input)

Exemplu:

APD:



$w = 01c10 \in L(A)$

$w = 023 \notin L(A)$

$w = 0010101c1010100 \in L(A)$

$w = 011c110 \notin L(A)$

2. Verificare generare cuvânt pentru o GIC

Input: o gramatică independentă de context oarecare și un cuvânt

Output: boolean (se precizează dacă acel cuvânt poate fi generat sau nu de gramatica dată)

Exemplu:

G:

$$N = \{S, A\}$$

$$T = \{a, b\}$$

$$S \rightarrow bSbb \mid A$$

$$A \rightarrow aA \mid \varepsilon$$

$$w = baaaabb \in L(G)$$

$$w = bbbbbb \in L(G)$$

$$w = aaaaaaa \in L(G)$$

$$w = bbbb \notin L(A)$$

$$w = c \notin L(A)$$

$$w = baaaa \notin L(A)$$

3. Pentru o GIC, sa se genereze toate cuvintele avand o lungime maxima data

Input: o gramatica independenta de context oarecare si un numar natural oarecare, n

Output: toate cuvintele de lungime mai mica sau egala cu n generate de gramatica data ca input

G:

$N = \{S, A\}$

$T = \{a, b\}$

$S \rightarrow bSbb \mid A$

$A \rightarrow aA \mid \varepsilon$

$n = 3$

Solutie: ε , bbb, a, aa , aaa

$n = 4$

Solutie: ε , bbb, a, aa , aaa, aaaa, babb

4. Transformare GIC in Forma Normala Chomsky

Input: O gramatica independenta de context oarecare

Output: O gramatica in FNC echivalenta cu gramatica data ca input

Exemplu:

G:

$$N = \{S, A, B\}$$

$$T = \{a, b\}$$

$$S \rightarrow ASA \mid aB$$

$$A \rightarrow B \mid S$$

$$B \rightarrow b \mid \varepsilon$$

$$S \rightarrow aS$$

$$S \rightarrow bA$$

$$A \rightarrow \varepsilon$$

$$A \rightarrow cA$$

Solutie:

$$N = \{S_0, S, A, B, X, Y\}$$

$$T = \{a, b\}$$

$$S_0 \rightarrow AX \mid YB \mid a \mid AS \mid SA$$

$$S \rightarrow AX \mid YB \mid a \mid AS \mid SA$$

$$A \rightarrow b \mid A \rightarrow b \mid AX \mid YB \mid a \mid AS \mid SA$$

$$B \rightarrow b$$

$$X \rightarrow SA$$

$$Y \rightarrow a$$

5. Algoritmul CYK (verificare generare cuvânt de către gramatică în FNC)

Input: O gramatică în FNC oarecare și un cuvânt

Output: matricea triunghiulară rezultată în urma aplicării algoritmului CYK + un răspuns boolean care indică dacă șirul dat ca input este acceptat de gramatică sau nu

Exemplu:

G:

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

$w = baaba$

Soluție:

$\{S, A, C\}$	$\leftarrow X_{1,5}$			
\emptyset	$\{S, A, C\}$			
\emptyset	$\{B\}$	$\{B\}$		
$\{S, A\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$	
$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
b	a	a	b	a

Cuvântul aparține limbajului.