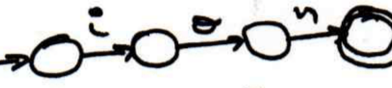
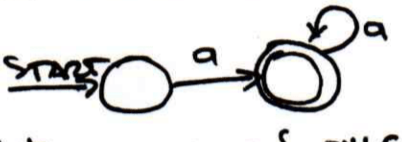
TEMA: GĂSIREA DIAGRAMEI DE STĂRI A UNUI AF și AF → AFD

CORESPUNZĂTOR "REȚETELOR" DE GHICIT GRAMATICI AVEM următoarele "AF elementare"

1) Lb. finit $L=\{ion\}$ $x_0 \to ion \in G_3$

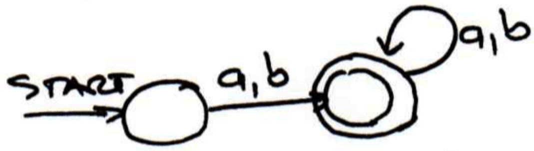


2) Repetarea unui simbol $L=\{a^n \mid n \geq 1\}$ $x_0 \to a x_0 \mid a \in G_3$



3) Amestec de litere $L=\{w \in \{a,b\}^* \mid w \neq \lambda\} \overset{not}{=} \{a,b\}^+$

$S \to aS \mid bS \mid a \mid b \in G_3$



Obs – rețeta 4) $S \to aSb \mid ab$ ce produce $\{a^n b^n \mid n \geq 1\}$ nu are corespondent deoarece e necesară gram. tip 2

FOLOSIREA "AF elementare" – Asemănător cu gramaticile

PAS1: Se identifică structura cuvântului, punând în evidență: poziții fixe, repetări de litere, amestec de litere

PAS2: Se identifică cel mai scurt cuvânt admis în limbaj și construim o structură liniară ce recunoaște cuvântul

PAS3: Se adaugă cicluri pt. repetări de litere sau amestecuri de litere

PAS4: Te bucuri dacă ai reușit! Algoritmul funcționează pt. multe cazuri, dar nu e general valabil (vom vedea folosind exemple)

Obs: 1) Construcția duce de obicei la AF nedet.

2). Pt. orice lb. de tip 3 există un șablon cu care se descrie orice cuvânt. Expresie regulată! Dar nu e așa trivial de găsit.

Exemplul 1: Găsiți un AF pth recunoașterea limbajului $L=\{a^n b^m c^k \mid m,n,k \geq 1\}$

Rezolvare

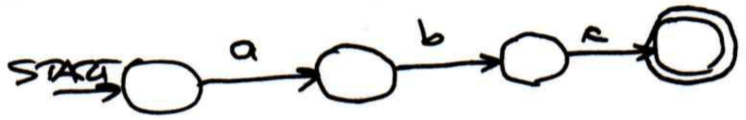
Structura cuvântului (șablonul)

$\underbrace{a \ldots a}_{n}\ \underbrace{b \ldots b}_{m}\ \underbrace{c \ldots c}_{k}$ $\qquad m,n,k \geq 1.$

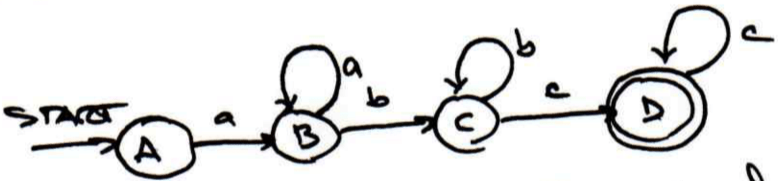
Cel mai scurt cuvânt din limbaj

$p = abc$ (m, n, k valori minime 1)

AF ce recunoaște p (rețeta 1)



Notăm stările (oricum!) și adăugăm repetori ale literelor (rețeta 2)



Descrierea construcției – după primul a ajung în starea B
și adaug repetori de a (ciclu pe starea B), apoi primul
b și ajung în starea C. Adaug repetori de litere b, ș.a.m.d.

La fel de corect ar fi

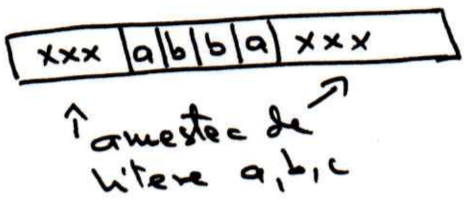
care corespunde unui AF nedeterminist (nu este simplu corect!). Găsiți și altele.

Exemplul 2: Găsiți un AF ptr. recunoașterea lb.

$L = \{ w \in \{a,b,c\}^* \mid w$ conține subcuvântul abba $\}$

Rezolvare

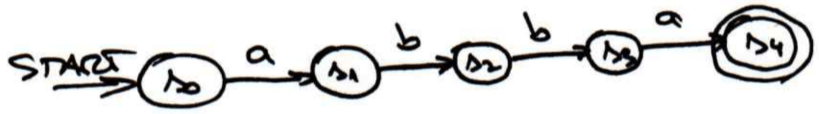
Structura cuvântului (undeva am șirul abba!)



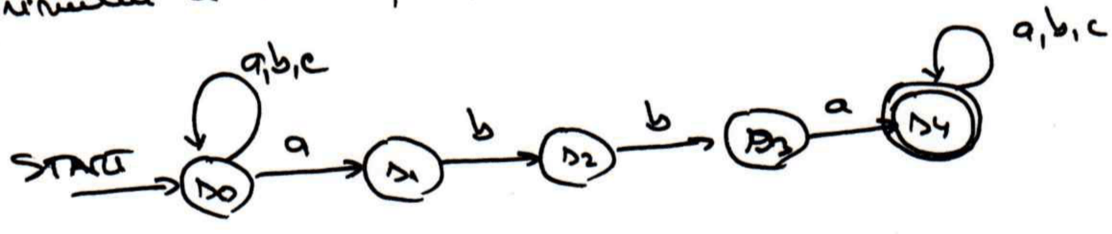
Cel mai scurt cuvânt din limbaj

$p = abba$

AF ce recunoaște p (rețeta 1)



Adaug amestecuri de litere înainte de citirea primului a din p și după ultimul a din p (rețeta 3)



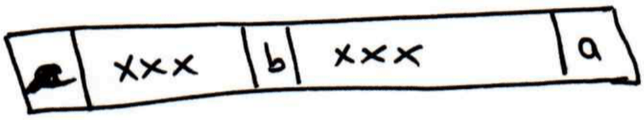
Obs: se obține un AF nedeterminist.

Exemplul 3 : Găsiți un AF pt. recunoașterea limbajului

$L = \{ w \in \{a,b,c\}^* \mid$ w începe cu litera c, se termină cu a și conține cel puțin un b $\}$

Rezolvare

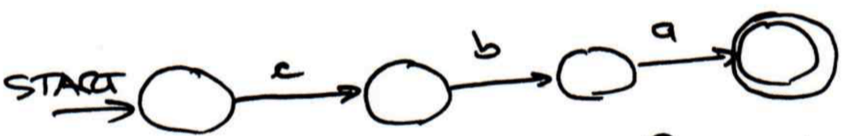
"Șablonul"



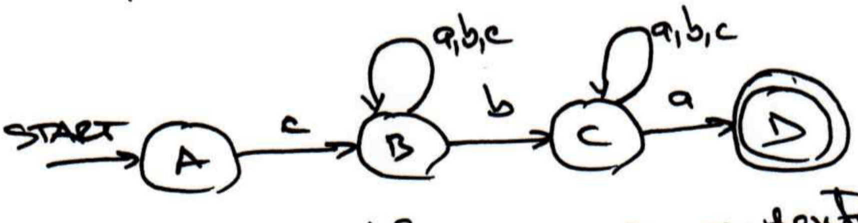
Cel mai scurt cuvânt

$p = cba$

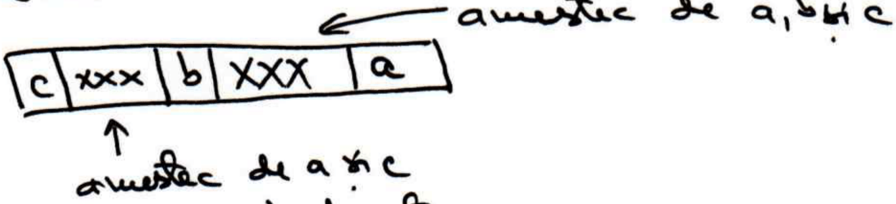
AF pt. rec. cuvântului p



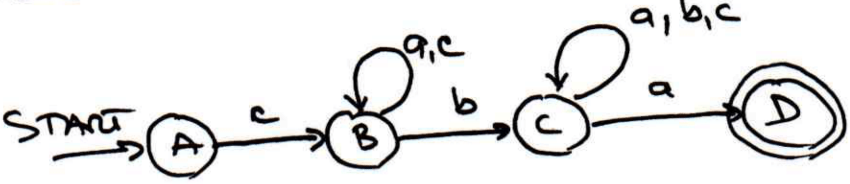
Adaug amestecuri în pozițiile potrivite



Obs: Dacă în șablon pun în evidență cea mai din stânga apariție a simbolului b atunci șablonul devine



iar AF echivalent este



Exemplul 4: Găsiți un AF ptr. recunoașterea lb

$L = \{ w \in \{0,1\}^* \mid w$ conține unul sau două simboluri $1 \}$.

Rezolvare

Problema: Avem 2 șabloane posibile pe care trebuie să le "combinăm"

Caz 1: un singur 1

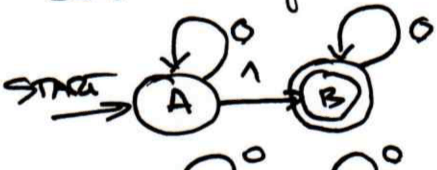
$\boxed{\underbrace{0\ldots0}_{k} \mid 1 \mid \underbrace{0\ldots0}_{t \text{ ori}}}$ $\quad k, t \geq 0$

Caz 2: două apariții de 1

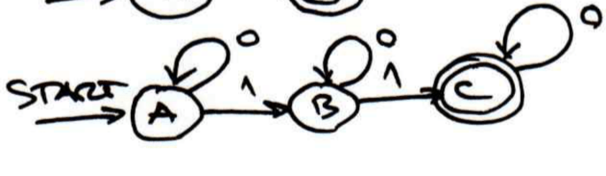
$\boxed{\underbrace{0\ldots0}_{k} \mid 1 \mid \underbrace{0\ldots0}_{m} \mid 1 \mid \underbrace{0\ldots0}_{n}}$ $\quad k, m, n \geq 0$

Pt. fiecare caz în parte am câte un AF

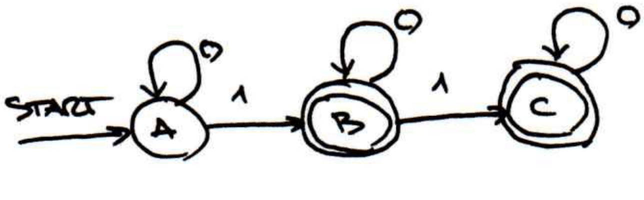
Caz 1:



Caz 2:



Combinarea AF – în general folosesc atâtea ramuri din starea inițială câte corespund șabloanelor, sau alte variante (de la caz la caz)



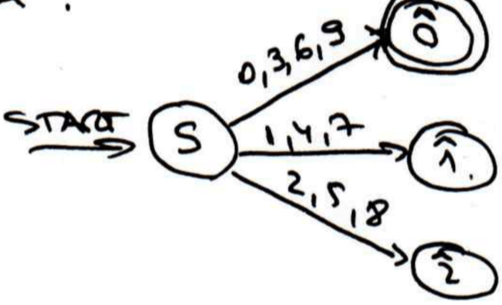
Exemplul 5: Construiți un AF pentru recunoașterea lb.

$$L = \{ w \in \{0, .., 9\}^+ \mid w \text{ multiplu de } 3 \}$$

Rezolvare

Deoarece nu avem un șablon (deși el există!) vom considera un automat finit în care asociem stările cu clasele de resturi modulo 3.

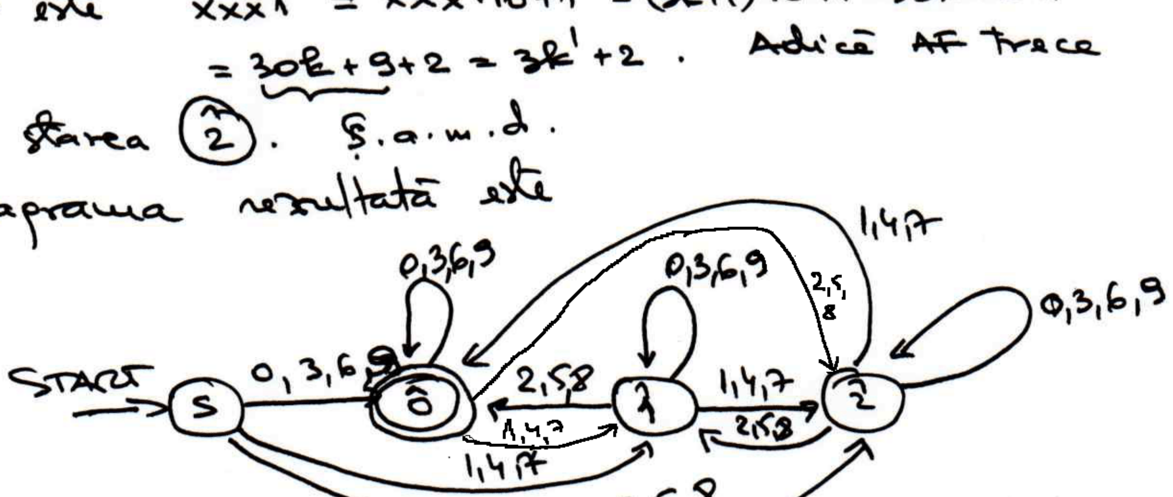
Deoarece se citește cel puțin o cifră vom considera o stare inițială din care se ajunge la stările corespunzătoare claselor prin citirea unei cifre.



Se analizează ce se întâmplă la citirea unei cifre dacă cuvântul examinat deja aparține unei clase de echivalență.

De exemplu, dacă șirul citit este $\overline{xxx}$ este în clasa $\hat{1}$, atunci $\overline{xxx} = 3k+1$ ptr. o valoare $k \in \mathbb{N}$. Citim o cifră, de ex. cifra 1. Noul număr citit este $\overline{xxx1} = \overline{xxx} \cdot 10 + 1 = (3k+1) \cdot 10 + 1 = 30k + 10 + 1 =$
$= \underbrace{30k + 9}_{} + 2 = 3k' + 2$. Adică AF trece în starea $\hat{2}$. ș.a.m.d.

Diagrama rezultată este



Exemplul 6: Găsiți un AF pt recunoașterea lb

$L = \{ w \in \{0,1,+,-\}^* \mid$ nr. întreg (binar) cu semn opțional $\}$

Rezolvare

Șabloane posibile

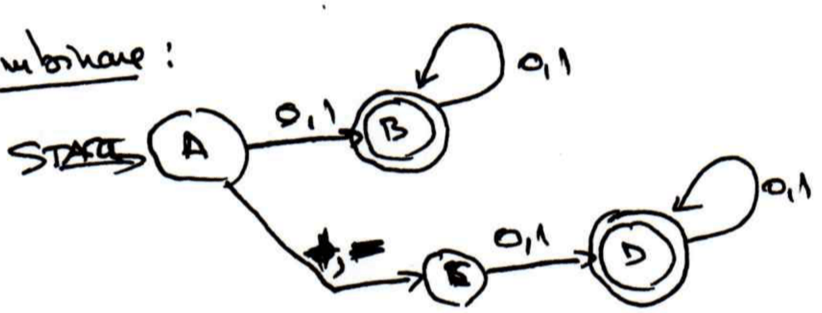
1) fără semn și cel puțin o cifră binară

$\boxed{XXX}$

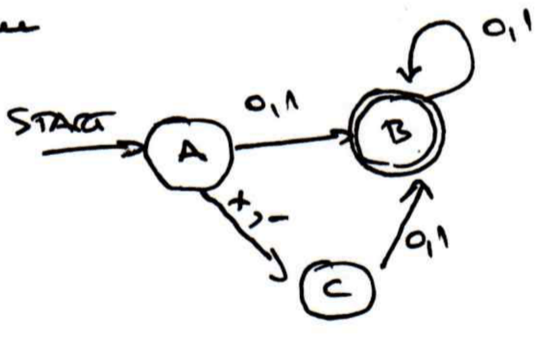
2) cu semn și cel puțin o cifră binară

$\boxed{\pm \mid XXX}$

Combinare:



Sau



Obs: Ambele automate sunt deterministe!

TRANSFORMAREA AFN → AFD simplu! (dar mult de lucru)

TEORIA:
$$AF = (S, I, f, s_0, S_f)$$
$$AFD = (\mathcal{P}(S), I, f, \{s_0\}, S_f') \quad \text{cu } S_f' = \{z \in \mathcal{P}(S) \mid z \cap S_f \neq \emptyset\}$$

Practic din starea $z \in \mathcal{P}(S)$ ajung în starea $z' \in \mathcal{P}(S)$ cu tranziția i dacă am o stare în AF (s) din care ajung în starea (s') la citirea lui i și $s' \in z'$.

Altfel opus, în AF consider toate arcele cu originea în stări din z și pun în z' toate destinațiile.

$$f(z, i) = \bigcup_{s \in z} f(s, i) \quad \text{propr. 3 de la extindere.}$$

AFD se construiește incremental, pornind cu $\{s_0\}$ și adăugând noi stări pt. fiecare literă citită. Adăugarea continuă pt fiecare stare nou adăugată până când nu mai apar stări noi.

Obs: Dacă se desenează inițial toate stările AFD (adică 2^n stări dacă $|S| = n$) atunci apar stări "neutilizabile", adică stări la care nu se poate ajunge când analizăm un cuvânt pornind cu starea inițială.

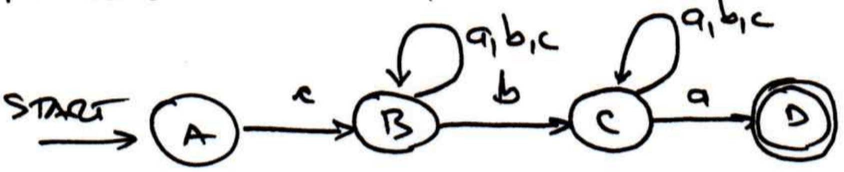
În general dacă pornim cu $|S| = n$ atunci în AFD avem un număr proporțional cu n de stări utile! $\Theta(n)$

Arcele ce corespund blocării → $\emptyset$ nu se mai desenează!

Exemplul 6: Reluăm cazul automatului ce recunoaște limbajul

$L = \{ w \in \{a,b,c\}^* \mid w$ începe cu litera c, se termină cu a și conține cel puțin un $b \}$

AF construit a fost

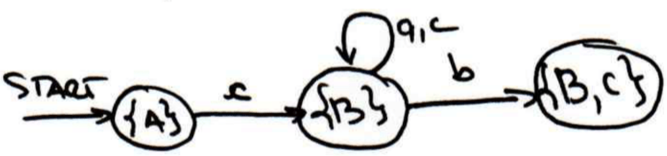


Transformăm AF în AFD, pornind cu $\{A\}$

A apărut $\{B\}$.

$f(\{B\},b) = f(B,b) = \{B,C\}$
$f(\{B\},a) = f(B,a) = \{B\}$
$f(\{B\},c) = f(B,c) = \{B\}$

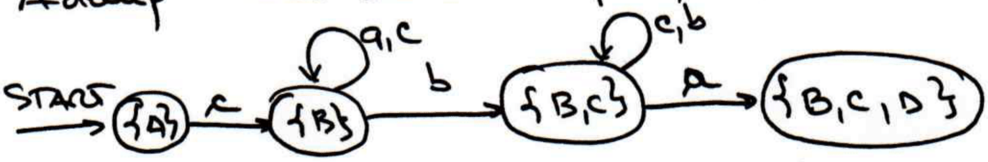
Adaug noile stări în graf.



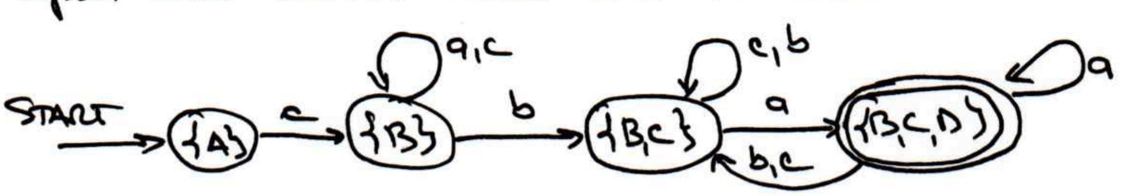
Analizez ieșirile din $\{B,C\}$, adică

$f(\{B,C\},a) = f(B,a) \cup f(C,a) = \{B\} \cup \{C,D\} = \{B,C,D\}$
$f(\{B,C\},c) = f(B,c) \cup f(C,c) = \{B\} \cup \{C\} = \{B,C\}$
$f(\{B,C\},b) = f(B,b) \cup f(C,b) = \{B,C\} \cup \{C\} = \{B,C\}$

Adaug noua stare în graf și reluăm procedeul



Se mai pot adăuga 2 arce dar nu mai apar alte stări. Deci am terminat



Exemplul 7: Găsiţi un AFD pentru recunoaşterea limbajului

$$L = \{ w \in \{0,1\}^* \mid w \text{ conţine cel puţin două simboluri } 1 \}.$$

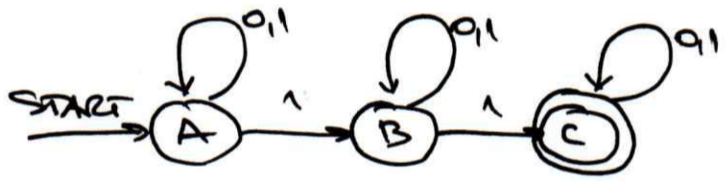
Rezolvare

Şablon uzual

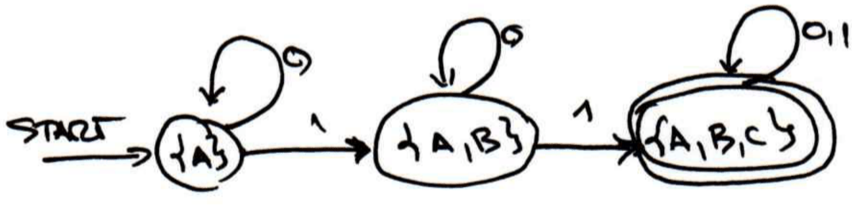
xxx	1	xxx	1	xxx

amestec de 0 şi 1.

AF



AFD



Obs: noul AFD corespunde şablonului în care se evidenţiază primele două apariţii ale lui 1 în cuvântul examinat.

0..0	1	0--0	1	xxx
k		j		

$k, j \geq 0$

Deci

$$L = \{ 0^k 1 0^j 1 w \mid k, j \geq 0,\ w \in \{0,1\}^* \}$$

este un alt mod de specificare a lui L adică am mai multe "şabloane" echivalente!

Exerciții propuse:

a) Construiți AF pentru recunoașterea următoarelor limbaje

1) $L = \{ w \in \{0,1\}^* \mid w$ are în penultima poziție cifra 1 $\}$

2) $L = \{ w \in \{a,b,c\}^* \mid w$ conține subcuvântul cc și are cel puțin patru litere $\}$

3) $L = \{ w \in \{0,1\}^* \mid w = 0^n 1^n, n \leq 3 \}$

4) $L = \{ w \in \{0,1\}^* \mid w$ multiplu de 4 $\}$

5) $L = \{ w \in \{0,1,..,9,/\}^* \mid w$ reprezintă o dată corectă într-un an bisect. Formatul datei este cel uzual ZZ/LL/AA $\}$

6) $L = \{ w \in \{0,.,9\}^* \mid w$ are 4 cifre și ultima este cifra de control, adică restul împărțirii sumei primelor 3 la 10 $\}$

b) Pentru AF construite la punctul a, găsiți AFD echivalente.