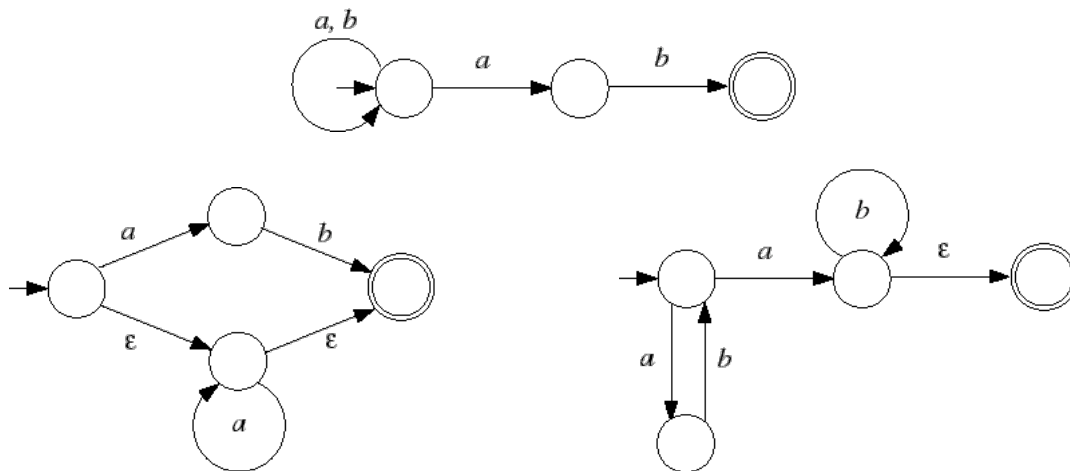


TD N° 1

1. Expressions régulières

a. Donner une expression pour les langages suivants :

- $\{a, b, aa, ab, ba, bb\}$
- les mots constitués d'un seul a , ou d'une chaîne de a (éventuellement vide) suivie d'un b
- les automates suivants



b. Comparer les langages définis par les expressions régulières, en indiquant les inclusions éventuelles :

- $(a|b)^*$ et $a^*|b^*$
- $(a|b)(c|d)$ et $ac|bd$
- a^{**} et a^* , que peut-on dire de x^{**} dans le cas général où x est une expression rationnelle ?
- $(b^*a)^*$ et $a^*(a|b)^*a$, comment peut-on caractériser ces deux langages ?
- $(a|b)^*$ et $(a^*b^*)^*$

2. Définitions régulières

Pour écrire des expressions régulières "en pratique" sous Unix, on utilise un certain nombre d'abréviations :

[abc] pour $a|b|c$

[a-z] pour $a|b|\dots|z$

[a-zA-Z0-9] pour toutes les lettres et chiffres

$x?$ pour un x optionnel

Il est également commode d'introduire des *définitions régulières*

$$d_1 = r_1$$

$$d_2 = r_2$$

...

$$d_n = r_n$$

où les d_i sont des identificateurs et les r_i des expressions régulières sur $\Sigma \cup \{d_1, \dots, d_{i-1}\}$.

Exemple : nombres = suites de chiffres ne commençant pas par 0

chiffre_non_nul = [1-9]

chiffre = chiffre_non_nul | 0

nombre_non_nul = chiffre_non_nul chiffre*

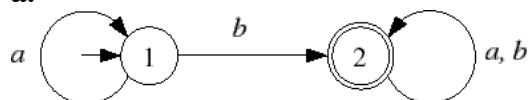
N.B. nombre_non_nul peut ensuite servir dans une autre définition (cf plus loin)

- Un identificateur est une suite de lettres ou de chiffres commençant par une lettre. Ecrire une définition régulière.
- Idem, mais en imposant la première lettre soit une majuscule (variables dans le langage de programmation Prolog)
- Un nombre entier divisible par 5.
- Chaines de lettres où les 5 voyelles apparaissent chacune une fois, et où elles sont dans l'ordre *a, e, i, o, u* (avec répétitions possibles)
- Nombres réels, tels que :
543 67.45 0.5 67.45E3 67.45E+3 67.45E-5

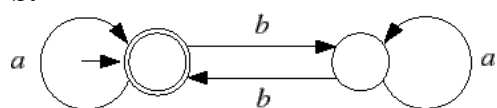
3. Décrire les langages reconnus par les automates suivants ($\Sigma = \{a,b\}$) ?

Construire les tables de transition des automates.

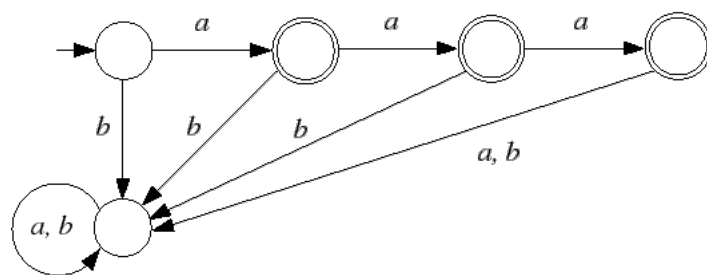
a.



b.



c.



4. Ecrire les automates reconnaissant les langages suivants

- $L_0 = \{\epsilon\}$, $L_0' =$ tous les mots non vides.
 - $L_2 = \{ab\}^*$ et $L_3 = \{ab\}^+$
 - $L_4 = \{ab, b\}^*$. Décrire la forme générale des mots de L_4 (« $L_4 =$ les mots qui... »).
 - $L_5 =$ le complémentaire de L_4 . Décrire la forme générale des mots de L_5 .
 - $L_6 = \{\text{mots se terminant par } aa \text{ et qui ont un nombre pair de } b\}$
- NB. Deux méthodes différentes : 1) comme produit d'automates 2) de manière directe

TD N° 1 - CORRECTION

1. $a | b | (a | b) (a | b)$

$a | a^*b$
 $(a | b)^*ab$
 $ab | a^*$
 $(ab)^*ab^*$

b. $(a | b)^*$ contient $a^* | b^*$

$(a | b) (c | d)$ contient $ac | bd$

$a^{**} = a^*$, et $x^{**} = x^*$ dans le cas général

$(b^*a)^* = a^* | (a | b)^*a$, tous les mots qui ne finissent pas par b

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

On a certaines égalités dans les expressions rationnelles :

$(a + b)^* = (a^*b)^*a^*$

$(ab)^* = \varepsilon + a(ba)^*b$

$a^* = \varepsilon + aa^*$, etc.

2. Définitions régulières

a. lettre = [a-zA-Z] chiffre=[0-9]

ident = lettre (lettre|chiffre)*

b. maj = [A-Z]

var = maj (lettre|chiffre)*

c. multiple_de_5 = nombre_non_nul? (0 | 5)

d. consonne = [b-df-hj-np-tv-z]

mot = (consonne* a consonne*)+ ... (consonne* u consonne*)+

e. entier = nombre_non_nul | 0

decimal = .chiffre* chiffre_non_nul

expt = E[+-]?entier

nb = entier decimal? expt?

3.

a. mots contenant au moins un b

b. nombre pair (éventuellement nul) de b

c. $\{a, aa, aaa\}$

4.

a. Il suffit d'un état pour reconnaître $\{\varepsilon\}$. Mais pour reconnaître le complémentaire, il faut compléter l'automate avec un état poubelle, puis intervertir les états finals et non finals. Le résultat est l'ensemble des mots avec au moins une lettre.