Automates, codes, graphes et matrices

MVA004

I. Gil-Michallon

CNAM

2010-2011

Chapitre 21 - Automates Finis

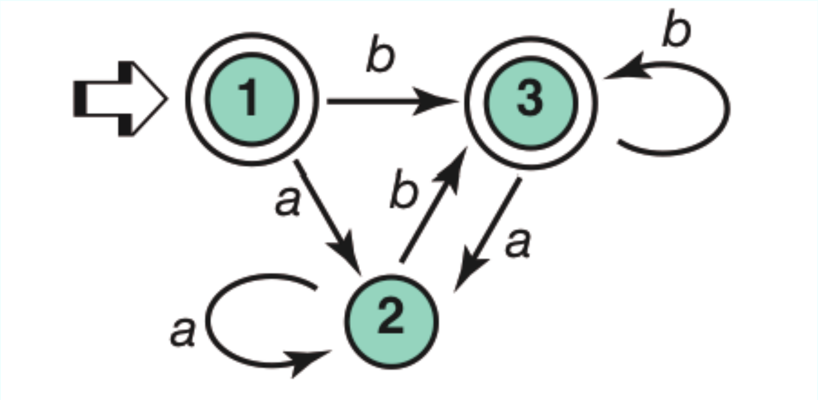
Langage d'un automate fini

- **Automate fini : A**

  Σ = alphabet
  $\mathcal{E}$ = ensemble d'états
  $\mathcal{A}$ = sous-ensemble d'états acceptants
  $\mathcal{I}$ = état initial
  $\delta : \mathcal{E} \times \Sigma \to \mathcal{E}$ = fonction de transition

L = Langage de l'automate = ensemble des mots reconnus par A
= ensemble des mots dont la lecture se termine sur un état acceptant

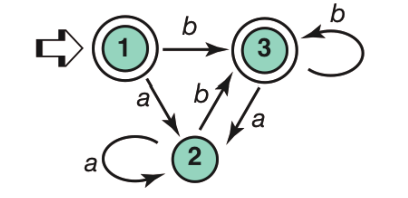
Dans un premier temps, on étudie les **AFD = Automates Finis Déterministes**


b
b
1
3
a
b
a
a
2

Méthode du départ

On associe à chaque état k un langage D_k = **langage de départ de l'état** k

D_k **= ensemble des mots acceptés si** k **était état initial**
L = D_1 **(si** 1 **est l'état initial)**



Mots de D_k **?**

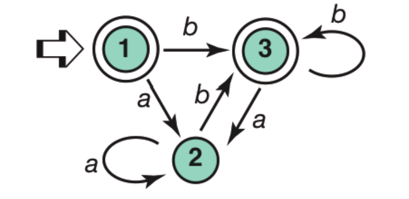
ϵ ?

mots de 1 lettre ?

mots qui commencent par ... ?

Système de Départ

$$\begin{cases} D_1 & = & \square & + & a_1 D_\square & + & a_2 D_\square & + & \cdots \\ D_2 & = & \square & + & a_1 D_\square & + & a_2 D_\square & + & \cdots \\ \cdots & & & & & & & & \\ D_n & = & \square & + & a_1 D_\square & + & a_2 D_\square & + & \cdots \end{cases}$$

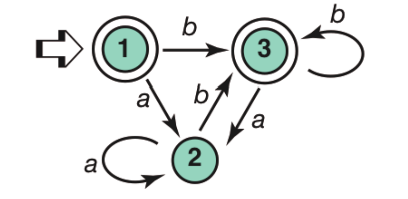


$$\begin{cases} D_1 & = & \square & + & aD_\square & + & bD_\square \\ D_2 & = & \square & + & aD_\square & + & bD_\square \\ D_3 & = & \square & + & aD_\square & + & bD_\square \end{cases}$$

Méthode de l'arrivée

On associe à chaque état k un langage A_k = **langage de l'arrivée de l'état k**

A_k **= ensemble des mots dont la lecture se termine sur l'état** k
L = somme des langages associés à des états acceptants



Mots de A_k ?

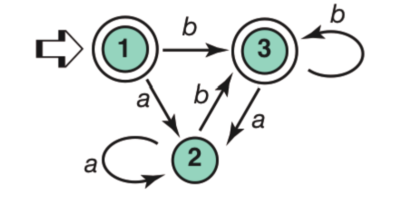
ϵ ?

mots de 1 lettre ?

mots qui se terminent par ... ?

Système de l'Arrivée

$$\left\{\begin{array}{lllllllll} A_1 & = & \square & + & A_\square a_\square & + & A_\square a_\square & + & \cdots \\ A_2 & = & \square & + & A_\square a_\square & + & A_\square a_\square & + & \cdots \\ \cdots & & & & & & & & \\ A_n & = & \square & + & A_\square a_\square & + & A_\square a_\square & + & \cdots \end{array}\right.$$



$$\left\{\begin{array}{lllllllll} A_1 & = & \square & & & & & & \\ A_2 & = & \square & + & A_\square a_\square & + & A_\square a_\square & + & A_\square a_\square \\ A_3 & = & \square & + & A_\square a_\square & + & A_\square a_\square & + & A_\square a_\square \end{array}\right.$$

Lemme de Arden

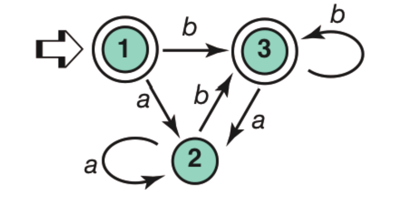
- **Si U est un langage ne contenant pas ϵ**
  l'équation $X = UX + V$ où X est un langage inconnu, admet comme unique solution
  $$X = U^* V$$
  L'équation $X = XU + V$ où X est un langage inconnu, admet comme unique solution
  $$X = VU^*$$
- **Si U est un langage contenant ϵ**
  l'équation $X = UX + V$ où X est un langage inconnu, admet comme solutions
  $$X = U^*(V + W)$$
  où W est un langage quelconque.
  L'équation $X = XU + V$ où X est un langage inconnu, admet comme solutions
  $$X = (V + W)U^*$$
  où W est un langage quelconque.

Méthode de résolution du système de départ

On choisit une des équations et on lui applique le Lemme de Arden, on obtient une expression pour D_k en fonction des autres langages de départ.

On remplace chacune des occurrences de D_k dans les autres équations du système.

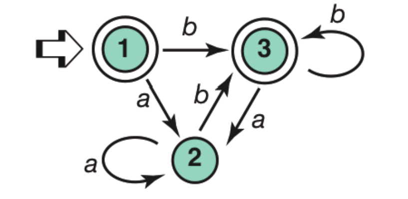
On recommence jusqu'à obtenir une expression régulière de D_1.



$$\begin{cases} D_1 & = & \epsilon & + & aD_2 & + & bD_3 \\ D_2 & = & & & aD_2 & + & bD_3 \\ D_3 & = & \epsilon & + & aD_2 & + & bD_3 \end{cases}$$

Méthode de résolution du système de l'arrivée

On choisit une des équations et on lui applique le Lemme de Arden, on obtient une expression pour A_k en fonction des autres langages d'arrivée.
On remplace chacune des occurrences de A_k dans les autres équations du système.
On recommence jusqu'à obtenir une expression régulière pour chacun des langages d'arrivée associés à un état acceptant. On en fait alors la somme.



$$\begin{cases} A_1 = \epsilon \\ A_2 = A_1 a + A_2 a + A_3 a \\ A_3 = A_1 b + A_2 b + A_3 b \end{cases}$$

Remarques sur les deux méthodes

- Selon l'ordre de résolution des équations, on n'obtient pas nécessairement les mêmes expressions régulières.
- Un automate admet un et un seul langage.
- Certains langages de départ ou d'arrivée sont très simples à déterminer :
  - si k est un piège acceptant : $D_k = \Sigma^*$
  - si k est un piège refusant : $D_k = \emptyset$
  - si k est inaccessible $A_k = \emptyset$
- Le langage de tout automate fini déterministe est régulier.