1
Les langages.

- **Un mot** est une chaîne de caractères.
- **Un langage** est un ensemble de mots.

Dans ce chapitre, on étudie des propriétés générales des langages.
Deux types de langages seront étudiés par la suite. Leur application à la compilation donne lieu à deux types d'analyse :

- **Les langages réguliers :** l'analyse lexicale,
- **Les langages algébriques :** l'analyse syntaxique.

1 – Les mots.

• Un *alphabet* est un ensemble $\mathcal{A}$, dont les éléments sont appelés *lettres*, *caractères* ou *symboles*.

Les mots

Un mot u sur l'alphabet $\mathcal{A}$ est une application

$$u : \{1, \ldots, m\} \to \mathcal{A}$$

où

> m est un entier appelé *la longueur* de u et noté $|u|$
> $\{1, \ldots, m\}$ est l'ensemble des entiers naturels i tels que $1 \leq i \leq m$.

• $u(i)$ est appelée *la* i*–ème lettre*, *le* i*–ème caractère* ou *le* i*–ème symbole* de u.

• Si $u(i) = x$, on dira que $u(i)$ *est une occurrence de* x *dans* u.

• On peut noter $u[i]$ au lieu de $u(i)$: on constate alors que la définition des mots nous est très familière !

L'ensemble des mots sur l'alphabet $\mathcal{A}$ est désigné par $\mathcal{A}^*$

1.1 – Définitions de base.

Soit $u \in \mathcal{A}^*$.

• Lorsque $|u| = 0$, $u : \emptyset \to \mathcal{A}$ est *le mot sans caractère* (*mot vide*), noté ε.

• Lorsque $|u| = 1$, $u : \{1\} \to \mathcal{A}$ est défini par le seul caractère $u(1) \in \mathcal{A}$.

On convient d'identifier tout $x \in \mathcal{A}$ au mot de longueur 1 qu'il définit :

$$\boxed{\mathcal{A} \subseteq \mathcal{A}^*}$$

Ces identifications ne sont valables que parce que les éléments de $\mathcal{A}$ sont reconnaissables pour tels.

• **L'adjonction d'une occurrence à droite** d'un mot $\mathcal{A}^* \times \mathcal{A} \to \mathcal{A}^*$ se définit de la façon suivante :

pour tout mot $u : \{1, \ldots, m\} \to \mathcal{A}$ et tout symbole $x \in \mathcal{A}$, $ux : \{1, \ldots, m+1\} \to \mathcal{A}$ est le mot défini par :

$$ux(i) = u(i) \text{ pour tout } i \in \{1, \ldots, m\},$$
$$ux(m+1) = x.$$

Remarques.

• $\varepsilon x = x$ si, comme il a été convenu ci–dessus, on identifie le symbole x avec le mot de longueur 1 qu'il définit.

• $|ux| = |u| + 1$.

1.2 – Récurrence sur les mots basée sur l'adjonction d'occurrences à droite.

Tout élément de $\mathcal{A}^*$ ou bien est ε ou bien s'obtient à partir d'un élément de $\mathcal{A}^*$ par "adjonction d'un caractère à droite". Les deux clauses suivantes :

- $\varepsilon \in \mathcal{A}^*$ (le mot sans caractère)
- pour tout $x \in \mathcal{A}$: si $u \in \mathcal{A}^*$ alors $ux \in \mathcal{A}^*$
  (adjonction de x à droite de u)

constituent donc une définition inductive de $\mathcal{A}^*$. Plus précisémént, $\mathcal{A}^*$ est le plus petit ensemble vérifiant les deux propriétés ci–dessus.

Exemple :

Construction	résultat	en abrégé
mot initial	ε	ε
adjonction de a	εa	a
adjonction de a	εaa	aa
adjonction de b	εaab	aab

Lorsque l'on écrit les mots par simple juxtaposition de caractères, il faut que ceux–ci n'interagissent pas les uns avec les autres : chaque caractère doit être indécomposable en des éléments appartenant à l'alphabet en cause.

La condition que $\mathcal{A}$ doit satisfaire pour cela s'exprime sous la forme suivante :

Lecture unique des mots sur $\mathcal{A}$

Quels que soient $u, v \in \mathcal{A}^*$, et $x, y \in \mathcal{A}$:

$$ux = vy \text{ implique } u = v \text{ et } x = y$$

Voici maintenant le principe qui est à la base de nombreuses démonstrations : sa justification découle directement de la construction de $\mathcal{A}^*$.

Récurrence sur les mots

Soit $P[u]$ un énoncé au sujet de $u \in \mathcal{A}^*$. Alors, si les deux propriétés suivantes sont vraies :

- $P[\varepsilon]$
- *pour tout* $x \in \mathcal{A}$ *et tout* $u \in \mathcal{A}^*$, $P[u]$ *implique* $P[ux]$

la propriété $P[u]$ est vraie pour tout $u \in \mathcal{A}^*$.

Ce principe n'est cependant pas le seul que l'on puisse utiliser pour raisonner sur les mots. En voici d'autres :

- construction des mots et récurrence par adjonction de caractères à gauche,
- induction sur la longueur des mots,

enfin, on sait qu'un ensemble non vide d'entiers contient un plus petit élément :

- un ensemble non vide de mots contient un mot dont la longueur est la plus petite possible.

Les notions de base relatives aux mots peuvent se définir par récurrence sur les mots : une telle définition est le **schéma d'une procédure récursive**.

- *La longueur* d'un mot se redéfinit par la récurrence :

$$|\varepsilon| = 0$$
$$|ux| = |u| + 1 \text{ pour tout } u \in \mathcal{A}^* \text{ et tout } x \in \mathcal{A}.$$

L'énoncé $P[u]$ est "$|u|$ est définie".

1.3 – La concaténation

est l'opération $\cdot\ : \mathcal{A}^* \times \mathcal{A}^* \rightarrow \mathcal{A}^*$ qui consiste à mettre deux mots "bout à bout".

La concaténation

Pour tout u et tout $v \in \mathcal{A}^*$, $u \cdot v$ est définie par récurrence sur v de la façon suivante :

1) $u \cdot \varepsilon = u,$ $\qquad I(u)$
2) $u \cdot (vx) = (u \cdot v)x$ $\qquad S(u, v, x)$
   pour tout $v \in \mathcal{A}^*$ et tout $x \in \mathcal{A}$.

L'énoncé $P[v]$ qui est en cause ici est "pour tout $u \in \mathcal{A}^*$, $u \cdot v$ est défini". La récurrence définissant la concaténation correspond à une procédure récursive :

$$\begin{aligned}
ab \cdot abba &= (ab \cdot abb)a && S(ab, abb, a) \\
&= ((ab \cdot ab)b)a && S(ab, ab, b) \\
&= (((ab \cdot a)b)b)a && S(ab, a, b) \\
&= ((((ab \cdot \varepsilon)a)b)b)a && S(ab, \varepsilon, a) \\
&= ((((ab)a)b)b)a && I(ab)
\end{aligned}$$

Propriétés de la concaténation

Quels que soient u, v et $w \in \mathcal{A}^*$:
0) $|u \cdot v| = |u| + |v|$.
1) Neutralité : $\varepsilon \cdot u = u$ et $u \cdot \varepsilon = u$.
2) Associativité : $(u \cdot v) \cdot w = u \cdot (v \cdot w)$.

On écrit très souvent uv au lieu de $u \cdot v$ et uvw au lieu de $(u \cdot v) \cdot w$ ou $u \cdot (v \cdot w)$.

Lemme de Lévy.

Si u, v, α et $\beta \in \mathcal{A}^*$ vérifient : $uv = \alpha\beta$ et $|v| \leq |\beta|$ (et donc aussi $|u| \geq |\alpha|$) alors il existe $\delta \in \mathcal{A}^*$ tel que $u = \alpha\delta$ et $\beta = \delta v$.

☞ La preuve se fait par récurrence sur v : l'énoncé en cause a la forme $P[v]$ = quels que soient $u, \alpha, \beta \in \mathcal{A}^*$ ....

- Si $v = \varepsilon$ alors $\delta = \beta$ convient parfaitement ;
- supposons que $P[v]$ soit vrai : il faut montrer qu'alors, $P[vx]$ est vrai pour tout $x \in \mathcal{A}$.
  Supposons donc que l'on ait $u(vx) = \alpha\beta$ et $|vx| \leq |\beta|$.
  Cette dernière condition implique $\beta \neq \varepsilon$ et on a donc $\beta' \in \mathcal{A}^*$ et $y \in \mathcal{A}$ tels que $\beta = \beta' y$.
  La première peut alors s'écrire (définition de la concaténation !) $(uv)x = (\alpha\beta')y$, ce qui (lecture unique des mots) implique $uv = \alpha\beta'$ et $y = x$;
  l'hypothèse de récurrence s'applique puisque l'on a aussi $|v| \leq |\beta'|$: on a donc un mot δ vérifiant $u = \alpha\delta$ et $\beta' = \delta v$, et donc $\beta = \beta' x = (\delta v)x = \delta(vx)$.

2 – Les langages, opérations élémentaires sur les langages.

- *Un langage* L sur $\mathcal{A}$ est une partie de $\mathcal{A}^*$: $L \subseteq \mathcal{A}^*$.
- L'ensemble des langages sur $\mathcal{A}$ est désigné par $\mathcal{P}(\mathcal{A}^*)$.

Exemples.

- $\emptyset \subseteq \mathcal{A}^*$, $\mathcal{A} \subseteq \mathcal{A}^*$ et $\mathcal{A}^* \subseteq \mathcal{A}^*$ sont des langages sur $\mathcal{A}$.
- Pour tout $u \in \mathcal{A}^*$, $\{u\} \subseteq \mathcal{A}^*$ est un langage sur $\mathcal{A}$.

On convient d'identifier un mot au langage à un seul élément qu'il définit : $\mathcal{A}^* \subseteq \mathcal{P}(\mathcal{A}^*)$.

Le regroupement de nos deux conventions vaut la peine d'être encadré :

$$\boxed{\mathcal{A} \subseteq \mathcal{A}^* \subseteq \mathcal{P}(\mathcal{A}^*)}$$

C'est généralement la notation la plus simple qui est utilisée, par exemple :

- si $x \in \mathcal{A}$, x désigne aussi le langage $\{x\}$ dont le seul mot ne comporte que le seul caractère x.
- ε désigne aussi le langage $\{\varepsilon\}$ dont le seul élément est le mot sans caractère.

2.1 – Sommes de langages = Réunions de langages.

L'ensemble des langages $\mathcal{P}(\mathcal{A}^*)$ étant l'ensemble des parties d'un ensemble, est muni d'opérations bien connues : **réunion**, intersection, complémentaire.

Soient $L \subseteq \mathcal{A}^*$ et $M \subseteq \mathcal{A}^*$ alors $L + M \subseteq \mathcal{A}^*$ désigne la réunion de L et M. Ceci revient à poser :

La somme

Pour tout $u \in \mathcal{A}^*$: $u \in L + M$ ssi $u \in L$ ou $u \in M$

Une caractérisation de la somme.

Quels que soient $L \subseteq \mathcal{A}^*$, $M \subseteq \mathcal{A}^*$ et $N \subseteq \mathcal{A}^*$:

$$\boxed{L + M \subseteq N \text{ ssi } L \subseteq N \text{ et } M \subseteq N}$$

Propriétés de la somme

Quels que soient $L \subseteq \mathcal{A}^*$, $M \subseteq \mathcal{A}^*$ et $N \subseteq \mathcal{A}^*$:

1) Neutralité : $L + \emptyset = L$ et $\emptyset + L = L$.
2) Associativité : $(L + M) + N = L + (M + N)$
   (la valeur commune s'écrit souvent $L + M + N$)
3) Commutativité : $L + M = M + L$.
4) Idempotence : $L + L = L$.
5) Croissance : $M \subseteq N$ implique $L + M \subseteq L + N$.

2.2 – Sommes généralisées de langages.

Une famille de langages sur $\mathcal{A}$, indexée par un ensemble I, est la donnée, pour chaque $i \in I$, d'un langage $L_i \subseteq \mathcal{A}^*$. Une famille indexée par un ensemble d'entiers (naturels) s'appelle généralement *une suite*.
Soit $(L_i)_{i\in I}$ une telle famille, alors $\sum_{i\in I} L_i \subseteq \mathcal{A}^*$ est **la réunion** des langages L_i.

Somme généralisée

Pour tout $u \in \mathcal{A}^*$:

$$u \in \sum_{i\in I} L_i \text{ ssi il existe } i \in I \text{ tel que } u \in L_i.$$

Cas particuliers.

$$\sum_{i\in\emptyset} L_i = \emptyset \qquad \sum_{i\in\{1\}} L_i = L_1 \qquad \sum_{i\in\{1,2\}} L_i = L_1 + L_2$$

I peut être un ensemble d'entiers naturels, un langage,...

Une caractérisation de la somme généralisée.

Pour tout $M \subseteq \mathcal{A}^*$:

$$\boxed{\sum_{i\in I} L_i \subseteq M \text{ ssi pout tout } i \in I,\ L_i \subseteq M}$$

2.3 – Concaténation des langages.

Soient L et $M \subseteq \mathcal{A}^*$ deux langages sur $\mathcal{A}$, alors tout élément de $L \cdot M \subseteq \mathcal{A}^*$ est obtenu en concaténant un élément de L et un élément de M, dans cet ordre.

Concaténation des langages

Pour tout $u \in \mathcal{A}^*$:

$$u \in L \cdot M \text{ ssi il existe } v \in L \text{ et } w \in M \text{ tels que } u = v \cdot w$$

Remarque.

Lorsque $L = v$ et $M = w$ sont des langages réduits à un seul élément, $L \cdot M = v \cdot w$ est un langage réduit à un seul élément : on a étendu aux langages la définition relative aux mots.

Propriétés de la concaténation

Quels que soient L, M et $N \subseteq \mathcal{A}^*$:

1) Neutralité : $L \cdot \varepsilon = L$ et $\varepsilon \cdot L = L$
2) Associativité : $(L \cdot M) \cdot N = L \cdot (M \cdot N)$
3) Croissance :
   $M \subseteq N$ implique $L \cdot M \subseteq L \cdot N$ et $M \cdot L \subseteq N \cdot L$
4) Nullité : $L \cdot \emptyset = \emptyset$ et $\emptyset \cdot L = \emptyset$
5) Distributivité : $L \cdot (M + N) = L \cdot M + L \cdot N$ et $(M + N) \cdot L = M \cdot L + N \cdot L$

Lorsque $M = \varepsilon$, la propriété 3) peut s'énoncer :

$$\varepsilon \in N \text{ implique } L \subseteq L \cdot N \text{ et } L \subseteq N \cdot L.$$

Les deux propriétés suivantes sont des cas particuliers de la distributivité de la concaténation par rapport aux sommes généralisées :

$$L \cdot \left(\sum_{i \in I} M_i \right) = \sum_{i \in I} (L \cdot M_i)$$

$$\left(\sum_{i \in I} M_i \right) \cdot L = \sum_{i \in I} (M_i \cdot L)$$

On écrit très souvent LM au lieu de $L \cdot M$ et LMN au lieu de $(L \cdot M) \cdot N$ ou $L \cdot (M \cdot N)$.

2.4 – Quelques définitions.

L*'ensemble* $fg(u) \subseteq \mathcal{A}^*$ *des facteurs gauches (préfixes) du mot* u, défini par la récurrence :

$$fg(\varepsilon) = \varepsilon$$
$$fg(ux) = fg(u) + ux \text{ pour tout } u \in \mathcal{A}^* \text{ et tout } x \in \mathcal{A}$$

vérifie :

$$v \in fg(u) \text{ ssi il existe } w \in \mathcal{A}^* \text{ tel que } u = vw$$

pour tout $u \in \mathcal{A}^*$ et tout $v \in \mathcal{A}^*$.

L*'ensemble* $fd(u) \subseteq \mathcal{A}^*$ *des facteurs droits (suffixes) du mot* u, défini par la récurrence :

$$fd(\varepsilon) = \varepsilon$$
$$fd(ux) = \varepsilon + fd(u)x \text{ pour tout } u \in \mathcal{A}^* \text{ et tout } x \in \mathcal{A}$$

vérifie :

$$v \in fd(u) \text{ ssi il existe } w \in \mathcal{A}^* \text{ tel que } u = wv$$

pour tout $u \in \mathcal{A}^*$ et tout $v \in \mathcal{A}^*$.

L*'ensemble* $fact(u) \subseteq \mathcal{A}^*$ *des facteurs du mot* u, défini par la récurrence :

$$fact(\varepsilon) = \varepsilon$$
$$fact(ux) = fact(u) + fd(u)x \text{ pour tout } u \in \mathcal{A}^* \text{ et tout } x \in \mathcal{A}$$

vérifie :

$$v \in fact(u) \text{ ssi il existe } w \in \mathcal{A}^* \text{ et } w' \in \mathcal{A}^* \text{ tels que } u = wvw'$$

pour tout $u \in \mathcal{A}^*$ et tout $v \in \mathcal{A}^*$.

3 – Itération des langages.

L'ensemble $\mathcal{A}^i$ des mots que l'on obtient à partir du mot sans caractère, par i adjonctions successives d'un caractère à droite, se définit par la récurrence suivante sur l'entier i

- $\mathcal{A}^0 = \varepsilon$ (un mot sans caractère)
- $\mathcal{A}^{i+1} = \mathcal{A}^i\mathcal{A}$ pour tout i.
  (on adjoint un caractère à droite, si $\mathcal{A} \neq \emptyset$)

$\mathcal{A}^*$ est l'ensemble de tous les mots ainsi construits :

$$\boxed{\mathcal{A}^* = \sum_{i \geq 0} \mathcal{A}^i}$$

Plus généralement, soit $L \subseteq \mathcal{A}^*$:

Le langage itéré d'un langage

- La puissance $L^i \subseteq \mathcal{A}^*$ de L est définie pour tout entier naturel i par la récurrence :
  - $L^0 = \varepsilon$
  - $L^{i+1} = L^i L$ pour tout i.
- Le langage itéré de L est le langage $L^* \subseteq \mathcal{A}^*$ défini par :

$$L^* = \sum_{i \geq 0} L^i.$$

Remarques.

- Cas où $L = a$ est réduit à un seul mot de longueur 1 :
  - a^i est réduit au mot de longueur i ne comportant que des occurrences du seul caractère a, par exemple $a^5 = aaaaa$;
  - $a^* = \{a^i \mid i \geq 0\}$ est l'ensemble des mots ne comportant que des occurrences du seul caractère a.

- Si $\mathcal{A} \neq \emptyset$, on peut vérifier par récurrence sur i que, pour tout $u \in \mathcal{A}^*$:
  - $u \in \mathcal{A}^i$ ssi $|u| = i$,
  - $u \in (\varepsilon + \mathcal{A})^i$ ssi $|u| \leq i$.

Propriétés des puissances

Quels que soient $L \subseteq \mathcal{A}^*$ et les entiers naturels i, j et k :

1) $L^1 = L$.
2) $L^i L^j = L^{i+j}$.
3) $L^i L = L L^i = L^{i+1}$.
4) $(\varepsilon + L)^k = \sum_{0 \leq i \leq k} L^i$.

Propriétés de l'itération

Quels que soient $L \subseteq \mathcal{A}^*$, $M \subseteq \mathcal{A}^*$ et $N \subseteq \mathcal{A}^*$:

1) Stabilité par concaténation : si $M \subseteq L^*$ et $N \subseteq L^*$ alors $MN \subseteq L^*$.
2) Stabilité par itération : si $M \subseteq L^*$ alors $M^* \subseteq L^*$.
3) Croissance : si $M \subseteq L$ alors $M^* \subseteq L^*$.
4) $\emptyset^* = \varepsilon$ et $\varepsilon^* = \varepsilon$.
5) $L^*L = LL^*$.
6) $L^*L^* = L^{**} = L^*$.
7) $L^* = \varepsilon + L^*L = \varepsilon + LL^*$.
8) $(L+M)^* = (L^* + M^*)^*$
$= (L^*M^*)^* = L^*(ML^*)^* = (L^*M)^*L^*$.

Remarque.

Quels que soient $L \subseteq \mathcal{A}^*$ et $M \subseteq \mathcal{A}^*$, on a :

$$\boxed{L^* \subseteq M \text{ ssi pour tout entier } i,\ L^i \subseteq M}$$

Lorsque l'on veut montrer une propriété du type $L^* \subseteq M$, il suffit de vérifier que $L^i \subseteq M$ pour tout i, ce qui peut se faire par récurrence sur i.

4 – Systèmes d'équations linéaires.

La notation multiplicative pour la concaténation et additive pour la réunion nous permet d'écrire des équations algébriques dans l'ensemble des langages sur un alphabet $\mathcal{A}$ et de tenter leur résolution.

4.1 – Equations linéaires à une inconnue.

Soient $A \subseteq \mathcal{A}^*$ et $B \subseteq \mathcal{A}^*$.

Une *solution de l'équation*

$$(E) \qquad X = AX + B$$

est un langage $L \subseteq \mathcal{A}^*$ vérifiant la relation $L = AL + B$.

Résolution de (E)

1) $L = A^*B$ est une solution de (E).
2) L est la plus petite solution de (E).
3) Si $\varepsilon \notin A$, alors (E) admet une solution unique.

---

2) signifie que si $M \subseteq \mathcal{A}^*$ vérifie $M = AM + B$, alors $A^*B \subseteq M$.

1) $AL + B = AA^*B + B = (AA^* + \varepsilon)B = A^*B = L$.

2) Soit M une solution de (E) : on a $M = AM + B$ donc, en particulier, $B \subseteq M$ et $AM \subseteq M$. Pour montrer $A^*B \subseteq M$, il suffit de vérifier que $A^iB \subseteq M$ pour tout i. C'est une récurrence facile :

   – pour $i = 0$, on a $A^0B = B \subseteq M$,
   – supposons que $A^iB \subseteq M$ alors
   $$A^{i+1}B = A(A^iB) \subseteq AM \subseteq M.$$

3) montrons d'abord, par récurrence sur k, que si M est une solution de (E) alors :
$$(E_k) \qquad M = A^{k+1}M + (\varepsilon + A)^kB$$
pour tout entier naturel k.

– (E_0) signifie simplement que M vérifie (E).
– Si (E_k) est vraie, alors, en utilisant $M = AM + B$, on peut écrire :
$$\begin{aligned} M &= A^{k+1}(AM + B) + (\varepsilon + A)^kB \\ &= A^{k+2}M + ((\varepsilon + A)^k + A^{k+1})B \\ &= A^{k+2}M + (\varepsilon + A)^{k+1}B \end{aligned}$$
ce qui implique (E_{k+1}).

Supposons que $\varepsilon \notin A$:

• pour tout $u \in \mathcal{A}^*$ on a $u \notin A^{|u|+1}M$, puisque les éléments de A sont au moins de longueur 1 ;

• si donc $u \in M$, $(E_{|u|})$ implique $u \in (\varepsilon + A)^{|u|}B \subseteq A^*B$: ceci signifie que $M \subseteq L$, c'est–à–dire $M = L$ par 2).

4.2 – Systèmes d'équations linéaires.

Soit m un entier naturel et soient $A_{i,j} \subseteq \mathcal{A}^*$ et $B_i \subseteq \mathcal{A}^*$ pour $i \leq m$ et $j \leq m$.
Une *solution du système*

$$(E) \quad X_i = \sum_{j \leq m} A_{i,j} X_j + B_i \qquad \text{pour } i \leq m$$

est un $(m+1)$–uplet $L = \begin{pmatrix} L_0 \\ \vdots \\ L_m \end{pmatrix}$ de langages sur $\mathcal{A}$ satisfaisant

$$L_i = \sum_{j \leq m} A_{i,j} L_j + B_i \text{ pour tout } i \leq m.$$

Les $(m+1)$–uplets de langages se comparent terme à terme, c'est–à–dire que

$$\begin{pmatrix} L_0 \\ \vdots \\ L_m \end{pmatrix} \subseteq \begin{pmatrix} M_0 \\ \vdots \\ M_m \end{pmatrix} \quad \text{ssi} \quad L_i \subseteq M_i \text{ pour tout } i \leq m.$$

Résolution de (E)

1) et 2) (E) admet une plus petite solution.
3) si $\varepsilon \notin A_{i,j}$ pour chaque i et chaque j, alors (E) admet une solution unique.

Méthode matricielle.

En utilisant la notation matricielle, on peut écrire (E) sous la forme $\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{B}$ où

$$\mathbf{X} = \begin{pmatrix} X_0 \\ \vdots \\ X_m \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} B_0 \\ \vdots \\ B_m \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} A_{0,0} & \dots & A_{0,m} \\ \vdots & \ddots & \vdots \\ A_{m,0} & \dots & A_{m,m} \end{pmatrix}$$

et montrer que sa plus petite solution est bien $\mathbf{A}^*\mathbf{B}$.

Ce résultat a un intérêt formel évident, mais, la manipulation de l'itérée $\mathbf{A}^*$ de la matrice carrée $\mathbf{A}$ n'est pas très aisé !

Méthode de Gauss.

La méthode de résolution adoptée ici est une transposition de la **méthode de Gauss**, basée sur l'appliction itérée de deux opérations élémentaires :

– **la résolution partielle,**
– **la substitution.**

• L'équation de X_i peut s'écrire $X_i = A_{i,i}X_i + C_i$ où C_i ne dépend pas de X_i : sous cette forme, il est possible de la "résoudre" en appliquant le résultat relatif à une équation unique ; on obtient ainsi la "résolvante partielle de (l'équation de) X_i" :

$$X_i = A^*_{i,i}C_i$$

dont le second membre ne dépend plus de X_i.

On vérifie que pour tout entier naturel $i \leq m$:

Résolution partielle

Soit (F) le système obtenu à partir de (E) en remplaçant l'équation de X_i par sa résolvante partielle, alors :
la plus petite solution de (F) est égale à la plus petite solution de (E).

• On vérifie que, pour tout couple d'entiers naturels $i \leq m$ et $j \leq m$ tels que $i \neq k$:

Substitution

Soit (F) le système obtenu à partir de (E) en remplaçant X_i par le second membre $\sum_{j \leq m} A_{i,j}X_j + B_i$ de son équation dans celle de X_k, alors :
la plus petite solution de (F) est égale à la plus petite solution de (E).

Exemple.

Considérons le système

$$\begin{array}{lcccccccc} X_0 & = & bX_0 & + & aX_1 & & & & \\ X_1 & = & & & & aX_2 & + & bX_3 & \\ X_2 & = & & & aX_1 & & + & bX_3 & + \ \varepsilon \\ X_3 & = & & & bX_1 & & + & aX_3 & \end{array}$$

Les opérations successives

- résolution de X_0,
- résolution de X_3,
- substitution de X_3 dans les équations de X_1 et X_2,
- substitution de X_2 dans l'équation de X_1

le transforment en :

$$\begin{array}{l} X_0 = b^*aX_1 \\ X_1 = (aa + ba^*b + aba^*b)X_1 + a \\ X_2 = (a + ba^*b)X_1 + \varepsilon \\ X_3 = a^*bX_1 \end{array}$$

Enfin, la résolution de X_1 et des substitutions, conduisent à la solution cherchée :

$$\begin{array}{l} X_0 = b^*a(aa + ba^*b + aba^*b)^*a \\ X_1 = (aa + ba^*b + aba^*b)^*a \\ X_2 = (a + ba^*b)(aa + ba^*b + aba^*b)^*a + \varepsilon \\ X_3 = a^*b(aa + ba^*b + aba^*b)^*a \end{array}$$

Remarque.

- L'expression de la plus petite solution peut varier très sensiblement en fonction de l'ordre suivant lequel on effectue les opérations.

5 – Monoïdes et morphismes de monoïdes.

On a fait des conventions qui, pour un alphabet $\mathcal{A}$, peuvent se résumer par

$$\boxed{\mathcal{A} \subseteq \mathcal{A}^* \subseteq \mathcal{P}(A^*)}$$

Plus explicitement, $\mathcal{A}$ désigne un alphabet et

- le mot de longueur 1 défini par $x \in \mathcal{A}$ est identifié à la lettre x ;
- le langage $\{u\}$ comportant $u \in \mathcal{A}^*$ comme unique élément est identifié au mot u.

De plus

- la concaténation est notée par une simple juxtaposition, évoquant une multiplication ;
- la réunion est notée par le sympbole d'addition ou le symbole somme.

Toutes ces conventions offrent une facilité d'écriture dont il ne faut pas se cacher les dangers. En particulier, **il faut toujours avoir à l'esprit la nature des objets que l'on manipule** en précisant la signification des "lettres" que l'on utilise : une lettre, en soit, n'a aucun sens !

5.1 – Définition des monoïdes.

Les *monoïdes* sont des ensembles munis d'une structure algébrique fort courante.

Les monoïdes

Un monoïde est un triplet $(D, \times, e)$ *où*

- D *est un ensemble,*
- $\times : D \times D \to D$ *est une opération binaire que l'on notera* $(x, y) \mapsto x \times y$,
- $e \in D$ *est un élément particulier de* D *;*

qui vérifie les propriétés suivantes :

- *Neutralité :* $x \times e = x$ *et* $e \times x = x$ *quel que soit* $x \in D$,
- *Associativité :* $(x \times y) \times z = x \times (y \times z)$ *quels que soient* $x \in D$, $y \in D$ *et* $z \in D$.

Exemples.

• L'ensemble $\mathbf{N}$ des entiers naturels est muni de deux structures de monoïdes : $(\mathbf{N}, +, 0)$ et $(\mathbf{N}, \times, 1)$.

• Les exemples qui suivent, où $L \subseteq \mathcal{A}^*$ est un langage sur $\mathcal{A}$, ne font que résumer quelques propriétés déjà vues :

- $(L^*, \cdot, \varepsilon)$, en particulier $(\mathcal{A}^*, \cdot, \varepsilon)$,
- $(\mathcal{P}(L^*), +, \emptyset)$, en particulier $(\mathcal{P}(\mathcal{A}^*), +, \emptyset)$,
- $(\mathcal{P}(L^*), \cdot, \varepsilon)$, en particulier $(\mathcal{P}(\mathcal{A}^*), \cdot, \varepsilon)$.

5.2 – Propriété principale de $\mathcal{A}^*$.

Lorsque l'on passe des ensembles aux applications, on est conduit à la définition suivante :

Morphismes de monoïdes

Soient $(D, \times, e)$ et $(D', \times', e')$ deux monoïdes.
Un morphisme $(D, \times, e) \to (D', \times', e')$ *est une application* $h : D \to D'$ *qui vérifie les propriétés suivantes :*

- $h(e) = e'$,
- $h(x \times y) = h(x) \times' h(y)$ *quels que soient* $x \in D$ *et* $y \in D$.

Par exemple, l'application “longueur” $u \mapsto |u|$ définit un morphisme $(\mathcal{A}^*, \cdot, \varepsilon) \to (\mathbf{N}, +, 0)$.

La plupart des morphismes de monoïdes provient de la construction suivante.

Propriété principale de $\mathcal{A}^*$

Soit $(D, \times, e)$ *un monoïde, alors*

toute application $f : \mathcal{A} \to D$ *s'étend de façon unique en un morphisme de monoïdes* $(\mathcal{A}^*, \cdot, \varepsilon) \to (D, \times, e)$.

Vérification de la propriété principale.

Si un tel morphisme $\overline{f}$ existe :

– "$\overline{f}$ étend f" signifie que $\overline{f}(x) = f(x)$ pour tout $x \in \mathcal{A}$;
– "$\overline{f}$ est un morphisme de monoïdes" implique $\overline{f}(\varepsilon) = e$ et, pour tout $u \in \mathcal{A}^*$ et tout $x \in \mathcal{A}$, $\overline{f}(ux) = \overline{f}(u) \times \overline{f}(x) = \overline{f}(u) \times f(x)$.

Or, les conditions

(1) $\overline{f}(\varepsilon) = e$
(2) $\overline{f}(ux) = \overline{f}(u) \times f(x)$ pour tout $u \in \mathcal{A}^*$ et tout $x \in \mathcal{A}$

définissent entièrement $\overline{f}$ par récurrence sur les mots sur $\mathcal{A}$.

L'application ainsi obtenue est un morphisme de monoïdes :

- on sait déjà que $\overline{f}(\varepsilon) = e$,
- il reste donc à vérifier que l'on a $\overline{f}(u \cdot v) = \overline{f}(u) \times \overline{f}(v)$ pour tout $u \in \mathcal{A}^*$ et tout $v \in \mathcal{A}^*$.

Ceci se fait par récurrence :

$$\begin{aligned}
\overline{f}(u \cdot \varepsilon) &= \overline{f}(u) && (u \cdot \varepsilon = u)\\
&= \overline{f}(u) \times e && (e \text{ est neutre pour } \times)\\
&= \overline{f}(u) \times \overline{f}(\varepsilon) && (\text{par (1) ci–dessus})
\end{aligned}$$

Supposons que $\overline{f}(u \cdot v) = \overline{f}(u) \times \overline{f}(v)$ et soit $x \in \mathcal{A}$:

$$\begin{aligned}
\overline{f}(u \cdot (vx)) &= \overline{f}((u \cdot v)x) && (u \cdot (ux) = (u \cdot v)x)\\
&= \overline{f}(u \cdot v) \times f(x) && (\text{par (2) ci–dessus})\\
&= (\overline{f}(u) \times \overline{f}(v)) \times f(x) && (\text{HR})\\
&= \overline{f}(u) \times (\overline{f}(v) \times f(x)) && (\text{associativité de } \times)\\
&= \overline{f}(u) \times \overline{f}(vx) && (\text{par (2) ci–dessus})
\end{aligned}$$

Intérêt de la propriété principale.

Dès que $\mathcal{A}$ n'est pas vide, $\mathcal{A}^*$ est infini, même lorsque $\mathcal{A}$ est fini, ce qui sera le cas dans toutes nos applications.

Vocabulaire et convention.

Nous dirons que l'application $\overline{f}$ ci–dessus est l'*extension de* f *aux mots* et, en nous inspirant de l'identification $\mathcal{A}$ à une partie de $\mathcal{A}^*$, nous la noterons simplement f : la propriété précédente nous le permet !

Application : les substitutions.

Soit $\mathcal{B}$ un second alphabet. L'extension aux mots d'une application $f : \mathcal{A} \to \mathcal{P}(\mathcal{B}^*)$, est un morphisme de monoïdes

$$(\mathcal{A}^*, \cdot, \varepsilon) \to (\mathcal{P}(\mathcal{B}^*), \cdot, \varepsilon).$$

Exemple.

Soit $\mathcal{A} = a + b$ un alphabet à deux lettres et soit f l'application définie par $f(a) = L$ et $f(b) = M$ où L et M sont deux langages sur $\mathcal{B}$, alors, par exemple

$$f(abaa) = LMLL.$$

Plus généralement, $f(u)$ est obtenu en remplaçant chaque caractère x de u par $f(x) \subseteq \mathcal{B}^*$.
Pour cette raison, une application du type $f : \mathcal{A} \to \mathcal{P}(\mathcal{B}^*)$ s'appelle souvent *une substitution*.

6 – Les substitutions.

Les substitutions jouent un grand rôle.

6.1 – Extension aux langages.

La somme (généralisée) est la réunion notée additivement :

pour tout ensemble I d'indices et toute famille $(L_i)_{i\in I}$ de langages sur $\mathcal{A}$, $\sum\limits_{i\in I} L_i$ désigne la somme des membres de cette famille, c'est–à–dire le langage sur $\mathcal{A}$ défini par

$$u \in \sum_{i\in I} L_i \text{ ssi il existe } i \in I \text{ tel que } u \in L_i.$$

Par exemple, tout langage est la somme de ses sous langages à un seul mot :

$$L = \sum_{u\in L} u.$$

Applications préservant les sommes

Une application $f : \mathcal{P}(A^*) \to \mathcal{P}(B^*)$ *préserve les sommes* ssi on a

$$f(\sum_{i\in I} L_i) = \sum_{i\in I} f(L_i)$$

pour toute famille $(L_i)_{i\in I}$ de langages sur $\mathcal{A}$.

Remarques et propriétés immédiates.

Soit $f : \mathcal{P}(A^*) \to \mathcal{P}(B^*)$ une application préservant les sommes.

- $f(L) = \sum_{u \in L} f(u)$ pour tout $L \subseteq \mathcal{A}^*$ puisque $L = \sum_{u \in L} u$.
- f préserve aussi les sommes finies :
  - $f(\emptyset) = \emptyset$
  - $f(L + M) = f(L) + f(M)$.
- f est croissante c'est–à–dire que pour tout L et tout $M \subseteq \mathcal{A}^*$, $L \subseteq M$ implique $f(L) \subseteq f(M)$.

Extension aux langages

Toute application $f : \mathcal{A}^* \to \mathcal{P}(B^*)$ s'étend de façon unique en une application $\mathcal{P}(A^*) \to \mathcal{P}(B^*)$ préservant les sommes.

En effet, supposons qu'une telle application $\overline{f}$ existe :

- "$\overline{f}$ étend f" signifie que $\overline{f}(u) = f(u)$ pour tout $u \in \mathcal{A}^*$;
- "$\overline{f}$ préserve les sommes" implique alors que $\overline{f}(L) = \sum_{u \in L} \overline{f}(u)$ pour tout $L \subseteq \mathcal{A}^*$.

Or, ceci définit entièrement $\overline{f}$ à partir de f par

$$\overline{f}(L) = \sum_{u \in L} f(u)$$

et on peut vérifier que l'application ainsi obtenue préserve les sommes.

Vocabulaire et convention.

L'application $\overline{f}$ ci–dessus est appelée l'*extension de f aux langages* et, est notée simplement f.

$$\boxed{v \in f(L) \text{ ssi il existe } u \in L \text{ tel que } v \in f(u).}$$

Application.

Soit $f : \mathcal{A} \to \mathcal{P}(\mathcal{B}^*)$ une substitution.

Par ce qui précède :

• f admet une extension unique aux mots
$$f : \mathcal{A}^* \to \mathcal{P}(\mathcal{B}^*)$$
qui est un morphisme de monoïdes
$$(\mathcal{A}^*, \cdot, \varepsilon) \to (\mathcal{P}(\mathcal{B}^*), \cdot, \varepsilon)).$$

• Cette dernière admet à son tour une extension unique aux langages
$$f : \mathcal{P}(\mathcal{A}^*) \to \mathcal{P}(\mathcal{B}^*)$$
cette extension est encore un morphisme de monoïdes
$$f : (\mathcal{P}(\mathcal{A}^*), \cdot, \varepsilon) \to (\mathcal{P}(\mathcal{B}^*), \cdot, \varepsilon).$$

Plus explicitement :

- $f(\varepsilon) = \varepsilon$,
- $f(LM) = f(L)f(M)$,

quels que soient $L \subseteq \mathcal{A}^*$ et $M \subseteq \mathcal{A}^*$.

En particulier :

- $f(L^i) = f(L)^i$ pour tout entier naturel i,
- $f(L^*) = f(L)^*$.

Exemple.

Une propriété comme $(L+M)^* = (L^*M^*)^*$ est assez délicate à démontrer directement lorsque L et M sont des langages quelconques.
Démontrer que l'on a $(a+b)^* = (a^*b^*)^*$ lorsque a et b sont des caractères, est beaucoup plus facile à concevoir et à écrire ; cette preuve est cependant suffisante pour obtenir le cas général : il suffit de considérer l'application $f : a+b \to \mathcal{P}(\mathcal{A}^*)$ telle que $f(a) = L$ et $f(b) = M$ et d'appliquer la substitution qu'elle définit aux deux membres de l'égalité précédente.

Résumons :

Extension d'une substitution aux langages

Toute substitution $f : \mathcal{A} \to \mathcal{P}(\mathcal{B}^*)$ *s'étend de façon unique en une application* $\overline{f} : \mathcal{P}(\mathcal{A}^*) \to \mathcal{P}(\ B^*)$ *qui vérifie les deux propriétés :*

- $\overline{f}$ *préserve les sommes généralisées,*
- $\overline{f}$ *est un morphisme de monoïdes*

$$(\mathcal{P}(\mathcal{A}^*), \cdot, \varepsilon) \to (\mathcal{P}(\mathcal{B}^*), \cdot, \varepsilon).$$

$\overline{f}$ est appelée l'*extension de f aux langages* et est le plus souvent notée f.

2. Les langages réguliers et les automates finis.

Les *langages réguliers* jouent un grand rôle dans la théorie des langages et la compilation.

Ils sont liés à deux notions :

– les *expressions régulières*, de nature algébrique, qui permettent de les définir,
– les *automates finis*, de nature géométrique, qui permettent de les reconnaître.

Dans toute la suite

les alphabets sont supposés finis

1 – Les langages réguliers.

Un langage régulier sur $\mathcal{A}$ est obtenu, à partir des ensembles finis de mots sur $\mathcal{A}$, par application d'un nombre fini d'*opérations régulières*, c'est–à–dire :

- réunions finies,
- concaténations,
- itérations.

Techniquement, il est intéressant de considérer un langage régulier comme l'interprétation d'une *expression régulière*; pour écrire ces expressions, nous utilisons les symboles suivants :

$\emptyset$, les $x \in \mathcal{A}$, $+$, $\cdot$, * et les parenthèses (et).

Les expressions régulières

$EReg(\mathcal{A})$ est l'ensemble des expressions définies par application des clauses inductives suivantes :

(a) $x \in EReg(\mathcal{A})$ pour tout $x \in \mathcal{A}$,

(b) $\emptyset \in EReg(\mathcal{A})$,

(c) si $\alpha \in EReg(\mathcal{A})$ et $\beta \in EReg(\mathcal{A})$ alors $(\alpha + \beta) \in EReg(\mathcal{A})$,

(d) si $\alpha \in EReg(\mathcal{A})$ et $\beta \in EReg(\mathcal{A})$ alors $(\alpha \cdot \beta) \in EReg(\mathcal{A})$,

(e) si $\alpha \in EReg(\mathcal{A})$ alors $\alpha^* \in EReg(\mathcal{A})$.

La première application de ce principe d'induction est la définition d'une interprétation

$$I : EReg(\mathcal{A}) \to \mathcal{P}(\mathcal{A}^*)$$

des expressions régulières par des langages.

I est définie par les clauses inductives suivantes :

(a) $I(x) = x$ pour tout $x \in \mathcal{A}$, (langage dont le seul élément est le mot à une seule lettre x)
(b) $I(\emptyset) = \emptyset$, (langage vide)
(c) $I((\alpha + \beta)) = I(\alpha) + I(\beta)$, (réunion de langages)
(d) $I((\alpha \cdot \beta)) = I(\alpha)I(\beta)$, (concaténation de langages)
(e) $I(\alpha^*) = I(\alpha)^*$. (itération d'un langage)

Les langages réguliers

$L \subseteq \mathcal{A}^*$ est un langage régulier sur $\mathcal{A}$ ssi

il existe $\alpha \in EReg(\mathcal{A})$ telle que $L = I(\alpha)$;

on dit alors que α est *une expression régulière de* L.

Remarques.

Il y a peu de différence entre une expression régulière et son interprétation : il n'est pas très dangereux de les confondre. Par exemple

$$(\alpha+\beta)^* = (\alpha^* + \beta^*)^*$$

veut dire :

$$(L+M)^* = (L^* + M^*)^*$$

où $L = I(\alpha)$ et $M = I(\beta)$.

- De même, on écrit $(ab + a^*)ba$ au lieu de

$$((((a \cdot b) + a^*) \cdot b) \cdot a) \text{ ou de } (((a \cdot b) + a^*) \cdot (b \cdot a)).$$

Une ER est, littéralement, **une présentation formelle d'un langage régulier.** Elle définit *la forme* (ou *le motif*) qui sert de modèle aux mots du langage en question.

Exemples de LR et d'ER.

- Le langage vide $\emptyset \subseteq \mathcal{A}^*$ est régulier.
- Le langage $\varepsilon \subseteq \mathcal{A}^*$, interprétation $\emptyset^*$ est régulier.
- Un langage réduit à un seul mot est régulier. En particulier tout $x \in \mathcal{A}$ est un langage régulier.
- Tout langage fini est régulier, en particulier $\mathcal{A}$ lui–même.
- $\mathcal{A}^* \subseteq \mathcal{A}^*$ est régulier.
- Pour toute ER α et tout entier n, on peut définir α^n par récurrence sur n.
- ER de quelques langages sur $\mathcal{A} = a + b$:
  - $\emptyset$, ε, a, b, $\mathcal{A} = a + b$ et $\mathcal{A}^* = (a + b)^*$ définissent des langages réguliers,
  - $\mathcal{A}^m = (a + b)^m$ est une ER de l'ensemble des mots de longueur m,
  - $(\varepsilon + \mathcal{A})^n = (\varepsilon + a + b)^n$ est une ER de l'ensemble des mots de longueur $\leq n$,
  - $\mathcal{A}^m(\varepsilon + \mathcal{A})^n = (a + b)^m(\varepsilon + a + b)^n$ est une ER de l'ensemble des mots dont la longueur est comprise entre m et $m + n$,
  - enfin, $((\varepsilon + a)b)^*(\varepsilon + a)$ est une ER du langage formé des mots qui ne comportent pas le facteur aa.
- Il existe des langages sur $\mathcal{A}$ qui ne sont pas réguliers, dès que $\mathcal{A} \neq \emptyset$.

L'ensemble des langages réguliers peut se caractériser d'une façon directe :

L'ensemble des langages réguliers

L'ensemble $Reg(\mathcal{A})$ des langages réguliers sur $\mathcal{A}$ est le plus petit qui vérifie :

1) $L \in Reg(\mathcal{A})$ pour tout **langage fini** $L \subseteq \mathcal{A}^*$,
2) si $L \in Reg(\mathcal{A})$ et $M \in Reg(\mathcal{A})$
alors $L + M \in Reg(\mathcal{A})$ et $LM \in Reg(\mathcal{A})$,
3) si $L \in Reg(\mathcal{A})$ alors $L^* \in Reg(\mathcal{A})$.

La réunion, la concaténation et l'itération des langages sont, pour cette raison, appelées **opérations régulières**.

1.1 – Equations linéaires à coefficients réguliers.

La plus petite solution d'un système d'équations linéaires se calcule à partir de ses coefficients, par application d'opérations régulières.

On a donc :

Propriété

La plus petite solution d'un système d'équations linéaires à coefficients réguliers est constituée de langages réguliers.

Le cas où les coefficients sont finis sera utilisé dans la démonstration du Théorème de Kleene.

2 – Automates déterministes et complets (ADC).

Action sur un ensemble

Soient Q un ensemble d'"états" et $\mathcal{A}$ un alphabet.
Une action de $\mathcal{A}$ sur Q est une application

$$\bullet : Q \times \mathcal{A} \to Q$$

que l'on note $(q, x) \mapsto q \bullet x$.

La propriété principale de $\mathcal{A}^*$ se traduit de la façon suivante :

Propriété principale

Une action $\bullet : Q \times \mathcal{A} \to Q$ s'étend de façon unique en une application

$$\bullet : Q \times \mathcal{A}^* \to Q$$

telle que, pour tout $q \in Q$

- $q \bullet \varepsilon = q$
- $q \bullet (ux) = (q \bullet u) \bullet x$ *pour tout $u \in \mathcal{A}^*$ et tout $x \in \mathcal{A}$.*

Cette construction est compatible avec la concaténation :

Soit $\bullet : Q \times \mathcal{A} \to Q$ une action de $\mathcal{A}$ sur Q, alors

$$q \bullet (uv) = (q \bullet u) \bullet v$$

pour tout $q \in Q$, tout $u \in \mathcal{A}^*$ et tout $v \in \mathcal{A}^*$.

Il y a deux représentations pratiques possibles d'une action :

- Une action $\bullet$ peut être présentée par *une table* sur laquelle on porte la valeur de $q \bullet x$ à l'intersection de la ligne $q \in Q$ et de la colonne $x \in \mathcal{A}$.
- Un *graphe de transition* est une représentation géométrique d'une action. Ce graphe est constitué de nœuds et d'arêtes étiquetés :
  - un nœud q pour tout $q \in Q$;
  - une arête $q \xrightarrow{x} r$ pour tout $q \in Q$, tout $x \in \mathcal{A}$ et $r = q \bullet x$.

Remarques.

• On ne précise pas l'orientation d'une arête dont l'origine et l'extrémité sont confondues.

• On colle plusieurs étiquettes sur une arête pour en représenter plusieurs de même origine et même extrémité, par exemple :

$$q \xrightarrow{x,y} r$$

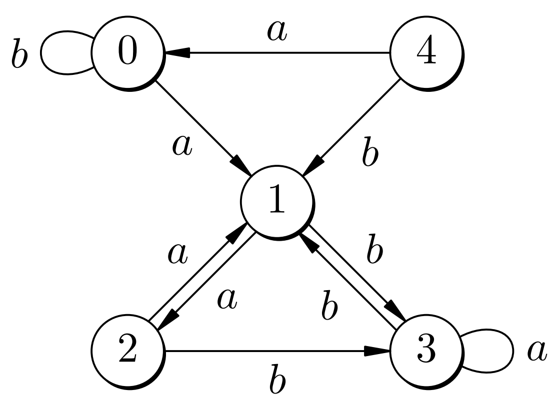
représente le couple d'arêtes

$$\begin{tikzcd} q \arrow[r, shift left, "x"] \arrow[r, shift right, "y"'] & r \end{tikzcd}.$$

Exemple 1.

$\mathcal{A} = a + b$, $Q = 0 + 1 + 2 + 3 + 4$

q	$q \bullet a$	$q \bullet b$
0	1	0
1	2	3
2	1	3
3	3	1
4	0	1



Chemins dans un graphe de transition.

Soit $q \in Q$ un état.

Chaque $u \in \mathcal{A}^*$ détermine un chemin unique $ch(q,u) \in Chem(q, q \bullet u)$

que l'on peut définir pour tout u par la récurrence suivante :

- $ch(q,\varepsilon) = q$;
- soit $ch(q,u) = \chi r$ avec $\chi \in Q^*$ et $r \in Q$, et soit $s = r \bullet x$ alors $ch(q,ux) = ch(q,u) \circ rs = ch(q,u)s$.

Dans l'exemple précédent, on a $ch(0, abab) = 01331$: la première lettre du mot sert à la fois à déterminer l'arête à parcourir et à payer ce parcours ; le mot se trouve raccourci et on peut ainsi poursuivre son chemin jusqu'à épuisement de ses lettres.

On complète cette image de "labyrinthe à péage" en adjoignant une entrée et des sorties :

Les ADC

Un automate déterministe et complet (en abrégé ADC) est la donnée d'un 5–uplet $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ où :

- Q est un ensemble (d'états),
- $\mathcal{A}$ est un alphabet fini,
- $\bullet : Q \times \mathcal{A} \rightarrow Q$ est une application (une action),
- $q_0 \in Q$ est l'entrée (ou état initial),
- $F \subseteq Q$ est l'ensemble des sorties (ou états finaux).

Exemple 1 (suite).

On peut compléter l'exemple 1, en choisissant

- l'entrée $q_0 = 0$
- la seule sortie $F = 2$.

Voici la table de cet ADC :

e/s	q	$q \bullet a$	$q \bullet b$
$\rightarrow$	0	1	0
	1	2	3
$\leftarrow$	2	1	3
	3	3	1
	4	0	1

Langage reconnu par un ADC

Le langage reconnu par un ADC $\mathbf{A}$ sur $\mathcal{A}$ est $\mathcal{L}(\mathbf{A}) \subseteq \mathcal{A}^*$ défini par :

$$u \in \mathcal{L}(\mathbf{A}) \text{ ssi } q_0 \bullet u \in F.$$

Deux ADC $\mathbf{A}$ et $\mathbf{A}'$ sont dits *équivalents* ssi

$$\mathcal{L}(\mathbf{A}) = \mathcal{L}(\mathbf{A}').$$

$\mathcal{L}(\mathbf{A})$ est donc l'ensemble des mots qui définissent des chemins partant de l'entrée q_0 et aboutissant à l'une des sorties.

3 – Automates finis déterministes et complets (AFDC).

Si l'on n'impose aucune condition supplémentaire, tout $L \subseteq \mathcal{A}^*$ est reconnu par un ADC approprié.

La condition qui est utile à notre problème est

l'ensemble Q des états est fini

et nous la supposons vérifiée dans toute la suite de ce chapitre : un ADC dont l'ensemble des états est fini est appelé un

automate fini déterministe et complet

ce que l'on abrègera en **AFDC**.

L'ADC présenté dans l'exemple 1 est évidemment un AFDC !

3.1 – Etats accessibles d'un AFDC.

Dans l'AFDC de l'exemple 1, aucun chemin partant de $q_0 = 0$ ne passera par l'état 4 :

Les états accessibles

$q \in Q$ est *accessible* ssi il existe $u \in \mathcal{A}^*$ tel que $q = q_0 \bullet u$.

Soit $\mathcal{A}cc$ l'ensemble des états accessible de $\mathbf{A}$: c'est le plus petit ensemble qui vérifie :

- $q_0 \in \mathcal{A}cc$,
- pour tout $q \in \mathcal{A}cc$ et tout $x \in \mathcal{A}$: $q \bullet x \in \mathcal{A}cc$.

Considérons l'AFDC $\mathbf{A}' = (Q', \mathcal{A}, \bullet, q_0, F')$ suivant :

- *Etats.* $Q' = \mathcal{A}cc$ (tous les états accessibles dans $\mathbf{A}$)
- *Entrée.* q_0 (état initial de $\mathbf{A}$)
- *Action.* $\bullet : Q' \times \mathcal{A} \to Q'$ (la restriction de $\bullet$ à Q')
- *Sortie.* $F' = F \cap Q'$ (sorties accessibles de $\mathbf{A}$)

Tous les états de $\mathbf{A}'$ sont accessibles à partir de q_0 et tout chemin partant de q_0 ne passe que par des états accessibles, on a donc $\mathcal{L}(\mathbf{A}') = \mathcal{L}(\mathbf{A})$.

Propriété

Tout AFDC est équivalent à un AFDC dont tous les états sont accessibles.

L'AFDC $\mathbf{A}'$ ainsi construit s'appelle

la partie accessible de $\mathbf{A}$.

C'est la seule utile lorsqu'on ne s'intéresse qu'à $\mathcal{L}(\mathbf{A})$.

Le calcul de $\mathcal{A}cc$ se fait en un nombre fini d'étapes !

• $\mathcal{A}cc$ se calcule de proche en proche comme la "limite" de la suite $\mathcal{U}_i$ suivante :

– $\mathcal{U}_0 = q_0$;
– $\mathcal{U}_{i+1} = \mathcal{U}_i + \mathcal{U}_i \bullet \mathcal{A}$.

La seconde clause, peut aussi s'écrire : pour tout $q \in Q$

$q \in \mathcal{U}_{i+1}$
ssi
$q \in \mathcal{U}_i$ ou il existe $r \in \mathcal{U}_i$ et $x \in \mathcal{A}$ tels que $q = r \bullet x$.

• $\mathcal{U}_i$, est une suite croissante de parties de l'ensemble fini Q, elle est donc stationnaire :

il existe $N < |Q|$ tel que $i \geq N$ implique $\mathcal{U}_i = \mathcal{U}_N$.

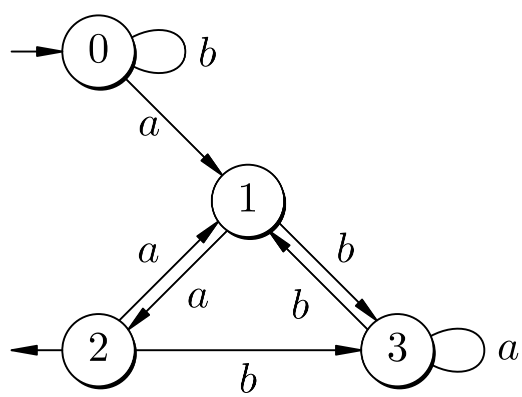
Il est clair que $\mathcal{U}_N$ est l'ensemble $\mathcal{A}cc$.

Exemple 1 (suite).

$$\mathcal{U}_0 = 0$$
$$\mathcal{U}_1 = 0 + 0 \bullet (a+b) = 0 + 0 \bullet a + 0 \bullet b = 0 + 1 + 0 = 0 + 1$$
$$\mathcal{U}_2 = 0 + 1 + (0+1) \bullet (a+b) = 0 + 1 + 2 + 3$$
$$\mathcal{U}_3 = 0 + 1 + 2 + 3 + (0+1+2+3) \bullet (a+b) = \mathcal{U}_2$$

On a donc $\mathcal{A}cc = 0 + 1 + 2 + 3$.

e/s	q	$q \bullet a$	$q \bullet b$
$\rightarrow$	0	1	0
	1	2	3
$\leftarrow$	2	1	3
	3	3	1



A titre d'exemple, voici un "programme" :

Calcul de $\mathcal{A}cc$

```
Acc := q0
tant qu'il existe q ∈ Acc non marqué faire
    sélectionner q ∈ Acc non marqué
    pour x parcourant 𝒜 faire
        Acc := Acc + q • x
    fin
    marquer q
fin
```

Le type qui convient à `Acc` est celui d'une file d'attente dont les éléments sont des couples `(état, marqué)` où `marqué` est un booléen.

Pour assurer la terminaison du programme, chaque élément n'entre dans la file et n'y est sélectionné qu'une fois au plus :

- le marquage signifie qu'un état a été traité complètement et qu'il ne sera jamais plus considéré,
- lorsque l'affectation `Acc := Acc + q • x` est exécutée, `Acc` n'est pas modifié si `q • x ∈ Acc`.

Remarque.

Le calcul de $\mathcal{A}cc$ est un algorithme qui permet de décider si $\mathcal{L}(\mathbf{A}) = \emptyset$ puisque cette propriété est équivalente à $\mathcal{A}cc \cap F = \emptyset$.

3.2 – Le théorème de Kleene.

Ce théorème fait le lien entre les langages réguliers et les langages reconnus par les AFDC : c'est le résultat essentiel de la théorie et c'est aussi la base des méthodes d'analyse lexicale.

THEOREME (Kleene)

Pour tout $L \subseteq \mathcal{A}^*$ les deux conditions suivantes sont équivalentes :

(a) L est régulier.

(b) Il existe un AFDC $\mathbf{A}$ tel que $L = \mathcal{L}(\mathbf{A})$.

La preuve du fait que (a) implique (b) consiste en la construction d'un AFDC à partir d'une expression régulière et sera faite plus tard.

Vérifions l'autre implication : soit $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ un AFDC, nous allons montrer que $\mathcal{L}(\mathbf{A})$ est l'une des composantes de la solution d'un système d'équations linéaires à coefficients réguliers (plus précisément, finis), satisfaisant la condition d'unicité : c'est donc un langage régulier.

Pour tout $q \in Q$ on considère le langage $Rec(q)$ reconnu par l'AFDC $\mathbf{A}(q) = (Q, \mathcal{A}, \bullet, q, F)$:

$$u \in Rec(q) \text{ ssi } q \bullet u \in F.$$

Propriétés.

Quels que soient $q \in Q$, $x \in \mathcal{A}$ et $u \in \mathcal{A}^*$

- $\varepsilon \in Rec(q)$ ssi $q \in F$;
- $xu \in Rec(q)$ ssi $xu \in xRec(q \bullet x)$.

☞ La première est évidente, pour la seconde :

$$\begin{array}{llr} xu \in Rec(q) & \text{ssi } q \bullet (xu) \in F & (\text{définition de } Rec(q)) \\ & \text{ssi } (q \bullet x) \bullet u \in F & (\text{propriété d'une action}) \\ & \text{ssi } u \in Rec(q \bullet x) & (\text{définition de } Rec(q \bullet x)) \\ & \text{ssi } xu \in xRec(q \bullet x) & \end{array}$$

En utilisant la construction des mots à partir du mot sans caractère par adjonction d'occurrences à gauche, ces propriétés permettent d'écrire :

$$Rec(q) = \begin{cases} \sum_{x \in \mathcal{A}} xRec(q \bullet x) + \varepsilon & \text{si } q \in F, \\ \sum_{x \in \mathcal{A}} xRec(q \bullet x) & \text{sinon.} \end{cases}$$

L'ensemble des égalités correspondant à tous les $q \in Q$ est un système d'équations linéaires dont les inconnues sont $X_q = Rec(q)$ et dont les coefficients sont réguliers (ε et des sommes finies d'éléments de $\mathcal{A}$) ; de plus la condition d'unicité est vérifiée :

$$\mathcal{L}(\mathbf{A}) = Rec(q_0) \text{ est donc un langage régulier.}$$

Exemple 1 (suite).

Le système d'équations correspondant à l'AFDC $\mathbf{A}'$ de l'exemple 1 précédent est :

$$\begin{array}{ccccccccccc}
X_0 & = & bX_0 & + & aX_1 & & & & & & \\
X_1 & = & & & & & aX_2 & + & bX_3 & & \\
X_2 & = & & & aX_1 & & & + & bX_3 & + & \varepsilon \\
X_3 & = & & & bX_1 & & & + & aX_3 & &
\end{array}$$

Ce système a été résolu dans le premier chapitre : la seule composante de la solution qui nous intéresse ici est X_0, qui est le langage reconnu par $\mathbf{A}'$:

$$\mathcal{L}(\mathbf{A}') = b^*a(aa + ba^*b + aba^*b)^*a.$$

4 – Automates finis (AF).

Pour démontrer la réciproque du théorème de Kleene, il faut construire un AF**DC** à partir d'une expression régulière :

- il est facile d'obtenir un objet satisfaisant, que l'on appelle un *automate fini* (AF en abrégé), représentable par un graphe de transition,
- il est bien rare qu'un AF soit un AF**DC** !

Un AF est défini par une *action non nécessairement déterministe et complète*, c'est–à–dire par une application

$$\bullet : Q \times \mathcal{A} \to \mathcal{P}(Q)$$

$q \bullet x$ n'est plus un état mais un ensemble d'états.

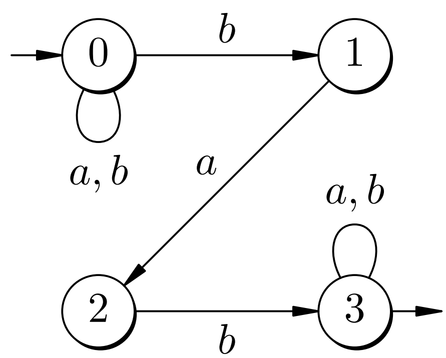
$q \bullet x$ peut donc :

- ou bien être vide : l'AF n'est alors *pas complet*,
- ou bien comporter plusieurs éléments : l'AF n'est alors *pas déterministe*.

Remarques.

• Les AF sont en général plus facile à concevoir que les AF**DC**.

Par exemple, l'AF représenté ci–dessous reconnaît de façon évidente l'ensemble des mots sur $\mathcal{A} = a + b$ comportant au moins un facteur *bab* :



mais sa table n'est pas très académique :

e/s	q	$q \bullet a$	$q \bullet b$
$\rightarrow$	0	0	$0 + 1$
	1	2	$\emptyset$
	2	$\emptyset$	3
$\leftarrow$	3	3	3

Les AF

Un automate fini (en abrégé AF) est la donnée d'un 5–uplet $\mathbf{A} = (Q, \mathcal{A}, \bullet, I, F)$ où :

- Q est un ensemble fini d'états
- $\mathcal{A}$ est un alphabet fini,
- $\bullet : Q \times \mathcal{A} \to \mathcal{P}(Q)$ est une application (une action non nécessairement DC)
- $I \subseteq Q$ est l'ensemble des entrées (ou états initiaux)
- $F \subseteq Q$ est l'ensemble des sorties (ou états finaux)

• La table définissant $\bullet : Q \times \mathcal{A} \to \mathcal{P}(Q)$ devra maintenant comporter des parties de l'ensemble Q.

On utilise encore les conventions faites au chapitre 1 :

- Tout ensemble à un seul élément $\{q\}$ est identifié à q lui–même :
$$Q \subseteq \mathcal{P}(Q)$$
- Les réunions sont notées additivement.

• On admet ici un ensemble d'entrées, au lieu d'une entrée unique.

• Les constituants du *graphe de transition* d'un AF sont les suivants :

- un nœud q pour tout $q \in Q$;
- une arête $q \xrightarrow{x} r$ pour tout $q \in Q$, tout $x \in \mathcal{A}$ et tout $r \in q \bullet x$.

4.1 – Dérivation dans un AF.

Soit $\mathbf{A} = (Q, \mathcal{A}, \bullet, I, F)$ un AF.

- *Une configuration* dans $\mathbf{A}$ est un couple $(q, u) \in Q \times \mathcal{A}^*$.
- *Une transition* est un changement de configuration en un seul pas : pour tout $q \in Q$ et tout $r \in Q$:

$$(q, xu) \underset{\mathbf{A}}{\overset{1}{\vdash}} (r, u) \text{ ssi } r \in q \bullet x$$

quel que soit $u \in \mathcal{A}^*$.

- *Une dérivation de longueur* i, $(q, u) \underset{\mathbf{A}}{\overset{i}{\vdash}} (r, v)$ est un enchaînement de i transitions :

$$0 : \ (q, u) \underset{\mathbf{A}}{\overset{0}{\vdash}} (q, u) \qquad \text{(pas de transition)}$$

$$i \mapsto i+1 : \ (q, u) \underset{\mathbf{A}}{\overset{i}{\vdash}} (s, w) \underset{\mathbf{A}}{\overset{1}{\vdash}} (r, v) \qquad \text{(une transition de plus)}$$

- On définit la relation $\underset{\mathbf{A}}{\overset{*}{\vdash}}$ entre configurations par :

$(q, u) \underset{\mathbf{A}}{\overset{*}{\vdash}} (r, v)$ ssi il existe une dérivation $(q, u) \underset{\mathbf{A}}{\overset{i}{\vdash}} (r, v)$.

Lorsqu'il n'y a pas d'ambiguïté sur l'AF $\mathbf{A}$ qui est en cause, on note $\overset{i}{\vdash}$ et $\overset{*}{\vdash}$ au lieu de $\underset{\mathbf{A}}{\overset{i}{\vdash}}$ et $\underset{\mathbf{A}}{\overset{*}{\vdash}}$.

AF particuliers.

AFD : Un AF est dit *déterministe* ssi
- I contient **au plus** un élément
- $q \bullet x$ contient **au plus** un élément

quels que soient $q \in Q$ et $x \in \mathcal{A}$.

AFC : Un AF est dit *complet* ssi
- I contient **au moins** un élément
- $q \bullet x$ contient **au moins** un élément

quels que soient $q \in Q$ et $x \in \mathcal{A}$.

Les AF qui sont à la fois déterministes et complets sont donc les AFDC. Un AF peut être un AFD et même un AFDC !

Langage reconnu par un AF

Le langage reconnu par un AF $\mathbf{A} = (Q, \mathcal{A}, \bullet, I, F)$ est $\mathcal{L}(\mathbf{A}) \subseteq \mathcal{A}^*$ défini par

$u \in \mathcal{L}(\mathbf{A})$

ssi

il existe $s \in I$ et $r \in F$ tels que $(s, u) \vdash_{\mathbf{A}}^{*} (r, \varepsilon)$.

Deux AF $\mathbf{A}$ et $\mathbf{A}'$ sont *équivalents* ssi $\mathcal{L}(\mathbf{A}) = \mathcal{L}(\mathbf{A}')$.

Dans un AF**DC**

$$r \in q \bullet x \text{ équivaut à } r = q \bullet x$$

quels que soient $q \in Q$, $r \in Q$ et $x \in \mathcal{A}$.

On voit alors que pour tout $q \in Q$ et tout $u \in \mathcal{A}^*$, il existe une dérivation unique $(q, u) \stackrel{i}{\longmapsto} (q \bullet u, \varepsilon)$, et que $i = |u|$.

Lorsqu'un AF $\mathbf{A}$ est déterministe et complet, la définition de $\mathcal{L}(\mathbf{A})$ est équivalente à celle donnée pour un AFDC.

Exemple 1 (suite).

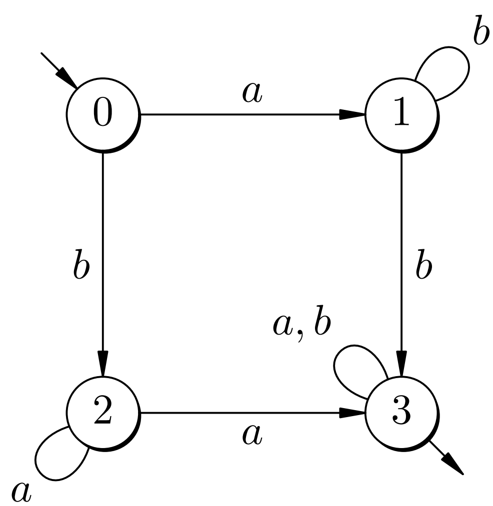
e/s	q	$q \bullet a$	$q \bullet b$
$\rightarrow$	0	1	0
	1	2	3
$\leftarrow$	2	1	3
	3	3	1
	4	0	1

$$\begin{aligned}
(0, abab) &\stackrel{1}{\longmapsto} (0 \bullet a, bab) = (1, bab) \\
&\stackrel{1}{\longmapsto} (1 \bullet b, ab) = (3, ab) \\
&\stackrel{1}{\longmapsto} (3 \bullet a, b) = (3, b) \\
&\stackrel{1}{\longmapsto} (3 \bullet b, \varepsilon) = (1, \varepsilon)
\end{aligned}$$

Exemple 2.

$\mathcal{A} = a + b$.

e/s	q	$q \bullet a$	$q \bullet b$
$\rightarrow$	0	1	2
	1	$\emptyset$	$1+3$
	2	$2+3$	$\emptyset$
$\leftarrow$	3	3	3



Dans cet AF, on a les dérivations

- $(0, abba) \overset{1}{\longmapsto} (1, bba) \overset{1}{\longmapsto} (1, ba) \overset{1}{\longmapsto} (3, a) \overset{1}{\longmapsto} (3, \varepsilon)$
- $(0, abba) \overset{1}{\longmapsto} (1, bba) \overset{1}{\longmapsto} (3, ba) \overset{1}{\longmapsto} (3, a) \overset{1}{\longmapsto} (3, \varepsilon)$

mais $(0, abba) \overset{1}{\longmapsto} (1, bba) \overset{1}{\longmapsto} (1, ba) \overset{1}{\longmapsto} (1, a)$ tombe dans une impasse puisque $1 \bullet a = \emptyset$.

4.2 – Détermination.

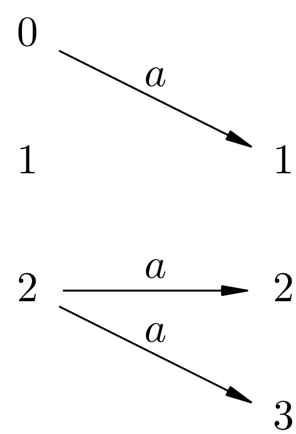
On peut simuler les tentatives pour construire un chemin de façon systématique en considérant l'extension

$$\bullet : \mathcal{P}(Q) \times \mathcal{A} \to \mathcal{P}(Q)$$

aux ensembles d'états de l'application $\bullet : Q \times \mathcal{A} \to \mathcal{P}(Q)$ définie par

$$S \bullet x = \sum_{s \in S} s \bullet x$$

pour toute $S \subseteq Q$ et tout $x \in \mathcal{A}$.



Exemple 2 (suite).

$$\begin{aligned}(0+1+2) \bullet a &= 0 \bullet a + 1 \bullet a + 2 \bullet a \\ &= 1 + \emptyset + (2+3) \\ &= 1 + 2 + 3\end{aligned}$$

Propriétés.

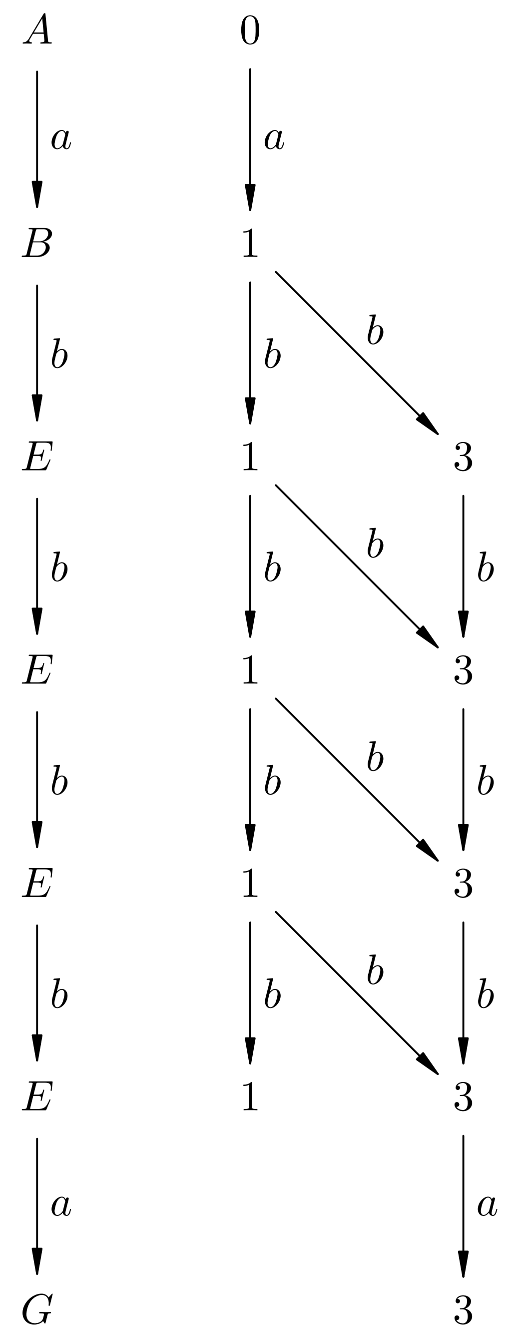
Soit $\mathbf{A} = (Q, \mathcal{A}, \bullet, I, F)$ *un AF alors, pour tout* $u \in \mathcal{A}^*$ *:*

1) *pour toute* $S \subseteq Q$ *et tout* $r \in Q$ *:*

   $r \in S \bullet u$ *ssi il existe* $s \in S$ *tel que* $(s, u) \vdash^{\!*} (r, \varepsilon)$,

2) $u \in \mathcal{L}(\mathbf{A})$ *ssi* $(I \bullet u) \cap F \neq \emptyset$.

Exemple 2 (suite).



La figure montre toutes les tentatives pour construire un chemin défini par le mot

$$u = abbbba$$

à partir de 0, dans l'AF de l'exemple 2 et, ce que l'on obtient en regroupant les états atteints, en le même nombre d'étapes.

Propriété de détermination

Tout AF est équivalent à un AFDC.

L'AFDC $DC(\mathbf{A})$ défini ci–dessous est équivalent à l'AF $\mathbf{A} = (Q, \mathcal{A}, \bullet, I, F)$: la construction précédente s'appelle la *détermination* (ou *déterminisation*) de $\mathbf{A}$.

$DC(\mathbf{A})$

$DC(\mathbf{A})$ est la partie accessible de l'AFDC $(Q', \mathcal{A}, \bullet, q_0', F')$ suivant :

- *Etats.* $S \in Q'$ ssi $S \subseteq Q$,
- *Action.* $\bullet : Q' \times \mathcal{A} \to Q'$, l'extension de $\bullet$ aux ensembles d'états,
- *Entrée.* $q_0' = I$,
- *Sorties.* $S \in F'$ ssi $S \cap F \neq \emptyset$.

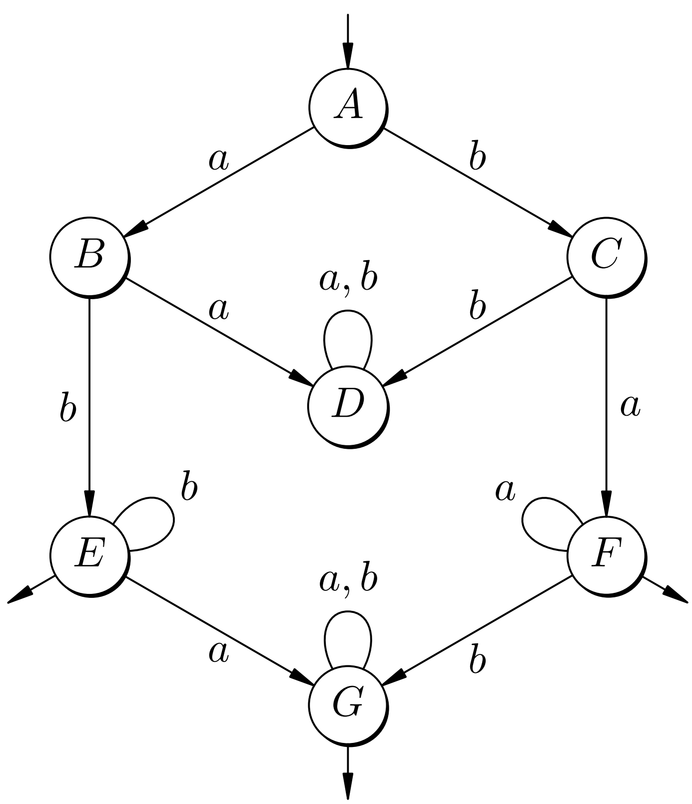
Le fait que $DC(\mathbf{A})$ est équivalent à $\mathbf{A}$ est exprimé par les propriétés ci–dessus.

Remarque.

Dans $DC(\mathbf{A})$ il peut se trouver des états non productifs, par exemple l'état vide : en les éliminant, on obtient un AFD noté $D(\mathbf{A})$, qui est excellent mais pas complet.

Exemple 2 (suite) sur la page suivante.

e/s	S	$S \bullet a$	$S \bullet b$
$\rightarrow$	$A = 0$	B	C
	$B = 1$	D	E
	$C = 2$	F	D
	$D = \emptyset$	D	D
$\leftarrow$	$E = 1 + 3$	G	E
$\leftarrow$	$F = 2 + 3$	F	G
$\leftarrow$	$G = 3$	G	G



4.3 – Les états productifs d'un AF.

La notion d'état accessible dans un AFDC s'adapte sans problème. Une notion duale est celle d'*état productif* : un état est productif s'il existe un chemin qui en part et qui mène à une sortie.

Les états productifs

$q \in Q$ est *productif* ssi il existe $u \in \mathcal{A}^*$ tel que

$$(q \bullet u) \cap F \neq \emptyset.$$

$(q \bullet u) \cap F \neq \emptyset$ signifie évidemment qu'il existe $r \in F$ tel que $(q, u) \overset{*}{\longmapsto} (r, \varepsilon)$.

L'ensemble $\mathcal{P}rod$ des états productifs de $\mathbf{A}$ est le plus petit ensemble qui vérifie :

- $F \subseteq \mathcal{P}rod$,
- pour tout $q \in Q$, s'il existe $x \in \mathcal{A}$ tel que
  $$(q \bullet x) \cap \mathcal{P}rod \neq \emptyset \text{ alors } q \in \mathcal{P}rod.$$

Ceci suffit pour construire l'AF $\mathbf{A}' = (Q', \circ, I', F)$ suivant :

- *Etats.* $Q' = \mathcal{P}rod$, (états productifs de $\mathbf{A}$)
- *Entrées.* $I' = I \cap \mathcal{P}rod$, (entrées productives)
- *Action.* $\circ : Q' \times \mathcal{A} \to Q'$, $(q \circ x = (q \bullet x) \cap \mathcal{P}rod)$
- *Sorties.* F.

Tous les états de $\mathbf{A}'$ sont productifs et, comme tout chemin aboutissant en F part d'un état productif, on a

$$\mathcal{L}(\mathbf{A}') = \mathcal{L}(\mathbf{A}).$$

Propriété

Tout AF est équivalent à un AF dont tous les états sont productifs.

Il est facile de vérifier que $\mathcal{P}rod$ est la limite de la suite croissante stationnaire $\mathcal{U}_i$ définie par la récurrence suivante :

- $\mathcal{U}_0 = F$,
- $\mathcal{U}_{i+1} = \mathcal{U}_i \bullet (\varepsilon + \mathcal{A})^{-1} = \mathcal{U}_i + \mathcal{U}_i \bullet \mathcal{A}^{-1}$.

On a utilisé l'extension aux langages de l'*action inverse* d'un élément de $\mathcal{A}$ sur un état, $q \bullet x^{-1} \subseteq Q$, définie par

$$r \in q \bullet x^{-1} \text{ ssi } q \in r \bullet x.$$

Cette construction permet de calculer effectivement $\mathcal{P}rod$ et d'imaginer un algorithme qui décide si $\mathcal{L}(\mathbf{A}) = \emptyset$, car cette condition équivaut à $I \cap \mathcal{P}rod = \emptyset$.

Attention.

Lorsque **A** est un AFDC, l'AF équivalent, dont tous les états sont productifs, est un AFD qui **n'est** généralement **pas complet**.

5 – Des AF encore moins déterministes.

Il est utile d'aller encore plus loin dans le non déterminisme en admettant des ε–transitions, c'est–à–dire des transitions gratuites !

ε–AF

Un ε–automate fini (en abrégé ε–AF) est la donnée d'un 5–uplet $\mathbf{A} = (Q, \mathcal{A}, \delta, I, F)$ où Q, $\mathcal{A}$, I et F sont comme précédemment, mais où

$$\delta : Q \times (\varepsilon + \mathcal{A}) \to \mathcal{P}(Q).$$

- Un AF est un ε–AF tel que $\delta(q, \varepsilon) = \emptyset$ pour tout $q \in Q$. En dehors de ce cas, la table définissant δ doit comporter une colonne pour les valeurs des $\delta(q, \varepsilon)$.

- Les constituants du *graphe de transition* d'un ε–AF sont les suivants :

  – un nœud q pour tout $q \in Q$;

  – une arête $q \xrightarrow{x} r$ pour tout $q \in Q$, tout $x \in \mathcal{A}$ et tout $r \in \delta(q, x)$.

5.1 – Dérivation dans un ε–AF.

Soit $\mathbf{A} = (Q, \mathcal{A}, \delta, q_0, F)$ un ε–AF :

• *Une transition* est un changement de configuration en un seul pas : pour $q \in Q$, $r \in Q$, $x \in \varepsilon + \mathcal{A}$ et $u \in \mathcal{A}^*$:

$$(q, xu) \underset{\mathbf{A}}{\overset{1}{\vdash\!\!-}} (r, u) \text{ ssi } r \in \delta(q, x)$$

pour un $u \in \mathcal{A}^*$.

On parle d'ε*–transition* lorsque $x = \varepsilon$ et de *transition sur* x lorsque $x \in \mathcal{A}$.

• *Une dérivation de longueur* i, notée $(q, u) \underset{\mathbf{A}}{\overset{i}{\vdash\!\!-}} (r, v)$ est un enchaînement de i transitions successives.

• *Une* ε*–dérivation* est une dérivation de la forme

$$(q, u) \underset{\mathbf{A}}{\overset{i}{\vdash\!\!-}} (r, u)$$

(si $i \neq 0$, c'est un enchaînement d'ε–transitions).

• On définit la relation $\underset{\mathbf{A}}{\overset{*}{\vdash\!\!-}}$ entre configurations par :

$$(q, u) \underset{\mathbf{A}}{\overset{*}{\vdash\!\!-}} (r, v) \text{ ssi il existe une dérivation } (q, u) \underset{\mathbf{A}}{\overset{i}{\vdash\!\!-}} (r, v).$$

Langage reconnu par un ε–AF

Le langage reconnu par un ε–AF $\mathbf{A} = (Q, \mathcal{A}, \delta, I, F)$ est $\mathcal{L}(\mathbf{A}) \subseteq \mathcal{A}^*$ défini par

$$u \in \mathcal{L}(\mathbf{A}) \text{ ssi il existe } s \in I \text{ et } r \in F \text{ tels que } (s, u) \underset{\mathbf{A}}{\overset{*}{\vdash\!\!-}} (r, \varepsilon).$$

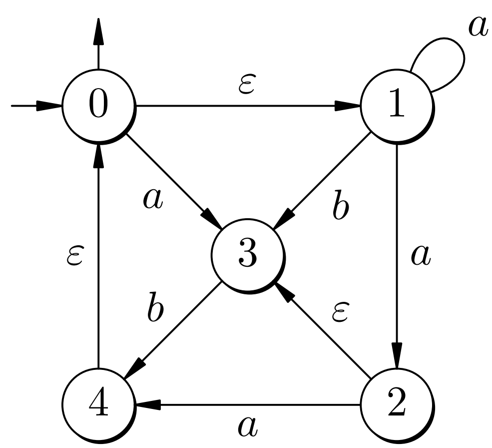
Deux ε–AF $\mathbf{A}$ et $\mathbf{A}'$ sont *équivalents* ssi $\mathcal{L}(\mathbf{A}) = \mathcal{L}(\mathbf{A}')$.

Exemple 3.

$\mathcal{A} = a + b$

$Q = 0 + 1 + 2 + 3 + 4.$

e/s	q	$\delta(q, \varepsilon)$	$\delta(q, a)$	$\delta(q, b)$
$\leftrightarrow$	0	1	3	$\emptyset$
	1	$\emptyset$	$1 + 2$	3
	2	3	4	$\emptyset$
	3	$\emptyset$	$\emptyset$	4
	4	0	$\emptyset$	$\emptyset$



5.2 – Détermination.

Les ε–AF savent reconnaître les mêmes langages que les AFDC.

Propriété de détermination

Tout ε–AF est équivalent à un AFDC.

Nous allons construire un AFDC $DC(\mathbf{A})$ équivalent à un ε–AF $\mathbf{A} = (Q, \mathcal{A}, \delta, I, F)$ donné : cette opération s'appelle la *détermination* (ou *déterminisation*) de $\mathbf{A}$.

Les états de $DC(\mathbf{A})$ sont des parties de Q qui, dans le cas où il existe effectivement des ε–transitions, ne peuvent pas être quelconques. En effet, si $\bullet$ désigne l'opération servant à définir l'action de $DC(\mathbf{A})$ et si $S \subseteq Q$ est un état de $DC(\mathbf{A})$, on devra avoir $S \bullet \varepsilon = S$; or, si $s \in S$, on doit certainement avoir $\delta(s, \varepsilon) \subseteq S$, ce qui signifie que S est close par ε–transition.

Clôture dans A.

La clôture de $q \in Q$ est l'ensemble $cl(q) \subseteq Q$ défini par :

$$r \in cl(q) \text{ ssi } (q, \varepsilon) \overset{*}{\longmapsto} (r, \varepsilon).$$

Cette définition s'étend à tout $S \subseteq Q$:

$$cl(S) = \sum_{s \in S} cl(s).$$

L'existence de la dérivation triviale $(q, \varepsilon) \overset{0}{\longmapsto} (q, \varepsilon)$ signifie que l'on a toujours $q \in cl(q)$.

Définition. $S \subseteq Q$ est dite *close* ssi $cl(S) = S$.

On peut remarquer que :

- $\emptyset$ est close,
- pour tout $q \in Q$ et toute $S \subseteq Q$ close : $q \in S$ ssi $cl(q) \subseteq S$,
- $cl(S)$ est close pour toute $S \subseteq Q$.

On étend δ à toute $S \subseteq Q$:

$$\delta(S, x) = \sum_{s \in S} \delta(s, x)$$

$DC(\mathbf{A})$

$DC(\mathbf{A})$ est la partie accessible de l'AFDC $(Q', \mathcal{A}, \bullet, q_0', F')$ suivant :

- *Etats.* $S \in Q'$ ssi $S \subseteq Q$ et S est close
- *Action.* $\bullet : Q' \times \mathcal{A} \to Q'$ est définie par
$$S \bullet x = cl(\delta(S, x))$$
- *Entrée.* $q_0' = cl(I)$
- *Sorties.* $S \in F'$ ssi $S \cap F \neq \emptyset$.

Exemple 3 (suite).

Pour calculer la table de $DC(\mathbf{A})$, il est commode de remplacer la colonne des valeurs de $\delta(q, \varepsilon)$ par celle des $cl(q)$ dans la table de $\mathbf{A}$.

e/s	q	$cl(q)$	$\delta(q,a)$	$\delta(q,b)$
$\leftrightarrow$	0	$0+1$	3	$\emptyset$
	1	1	$1+2$	3
	2	$2+3$	4	$\emptyset$
	3	3	$\emptyset$	4
	4	$0+1+4$	$\emptyset$	$\emptyset$

Considérons $B = 1+2+3$.

- B est clos car :

$$\begin{aligned} cl(1) &= 1 \subseteq B \\ cl(2) &= 2+3 \subseteq B \\ cl(3) &= 3 \subseteq B \end{aligned}$$

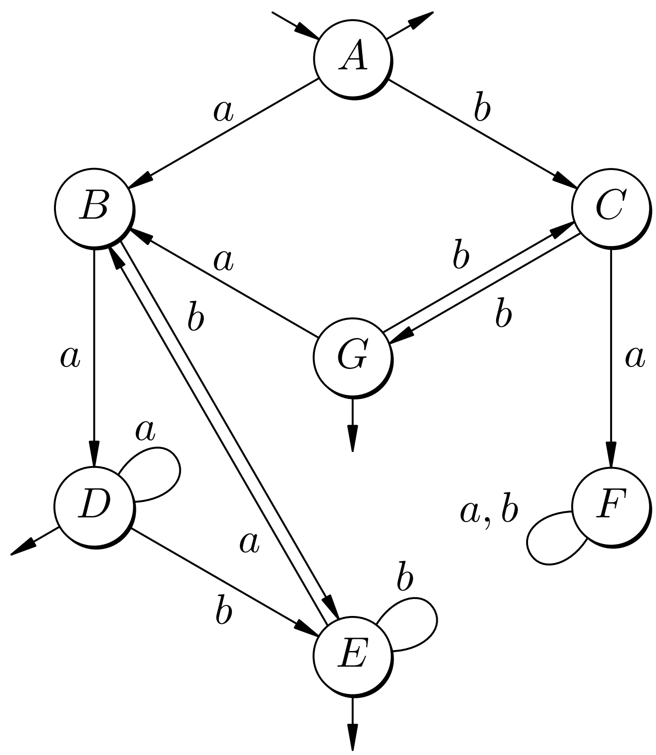
donc $cl(B) \subseteq B$.

- Voici les actions sur B :

$$\begin{aligned} B \bullet a &= cl(1+2) + cl(4) + \emptyset = 0+1+2+3+4 \\ B \bullet b &= cl(3) + \emptyset + cl(4) = 0+1+3+4. \end{aligned}$$

Le calcul de $DC(\mathbf{A})$ se fait, de proche en proche, en partant de $A = cl(0)$: on ne s'intéresse qu'à la partie accessible !

e/s	S	$S \bullet a$	$S \bullet b$
$\leftrightarrow$	$A = 0 + 1$	B	C
	$B = 1 + 2 + 3$	D	E
	$C = 3$	F	G
$\leftarrow$	$D = 0 + 1 + 2 + 3 + 4$	D	E
$\leftarrow$	$E = 0 + 1 + 3 + 4$	B	E
	$F = \emptyset$	F	F
$\leftarrow$	$G = 0 + 1 + 4$	B	C



5.3 – La propriété de détermination.

La propriété $\mathcal{L}(DC(\mathbf{A})) = \mathcal{L}(\mathbf{A})$ n'est pas parfaitement évidente bien qu'elle soit très analogue à celle du cas des AF.

Propriétés.

Soient $\mathbf{A} = (Q, \mathcal{A}, \delta, I, F)$ *un* ε*–AF et* Q' *l'ensemble des parties closes de* Q *alors, pour tout* $u \in \mathcal{A}^*$ *:*

1) *quels que soient* $S \in Q'$, $x \in \mathcal{A}$ *et* $r \in Q$ *:*

$r \in S \bullet u$

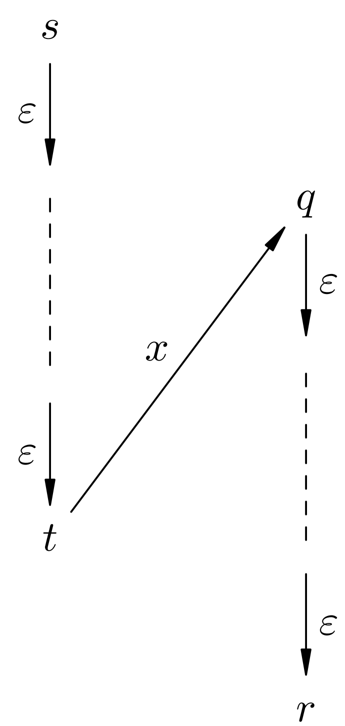
ssi

il existe $s \in S$ *tel que* $(s, u) \overset{*}{\longmapsto} (r, \varepsilon)$,

2) $u \in \mathcal{L}(\mathbf{A})$

ssi

$(cl(I) \bullet u) \cap F \neq \emptyset$.

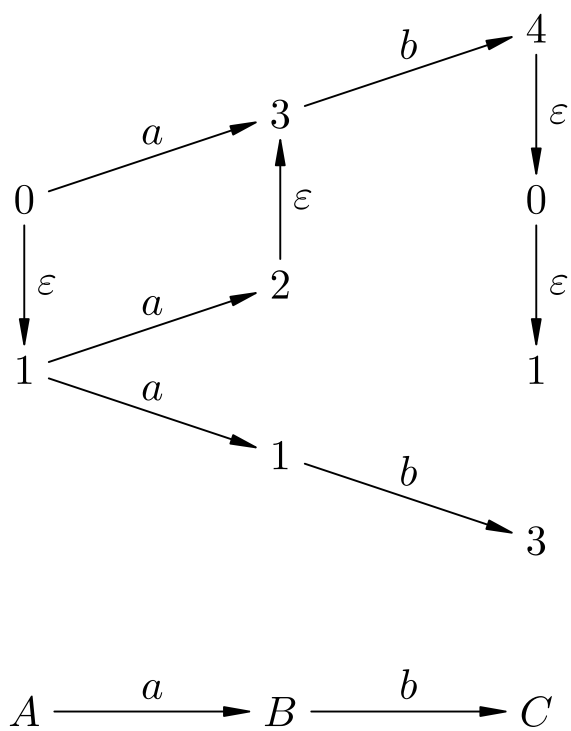


La figure ci–contre illustre 1) pour $u = x \in \mathcal{A}$.

$$S \xrightarrow{x} S \bullet x$$

Exemple 3 (suite).

La figure montre toutes les tentatives pour construire un chemin défini par le mot $u = ab$ à partir de l'état 0 dans l'ε–AF de l'exemple 3, et, les mêmes tentatives dans l'AFDC équivalent.



5.4 – Fin de la démonstration du théorème de Kleene.

Il nous reste à démontrer que (a) implique (b) dans le théorème :

THEOREME (Kleene).
Pour tout $L \subseteq \mathcal{A}^*$ *les deux conditions suivantes sont équivalentes :*

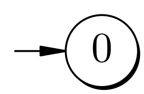
(a) L est régulier ;
(b) il existe un AFDC $\mathbf{A}$ *tel que* $L = \mathcal{L}(\mathbf{A})$*.*

Il faut construire un AFDC reconnaissant L pour chaque langage régulier L : nous construisons un ε–AF (à une seule entrée) $\mathbf{A} = (Q, \mathcal{A}, \delta, q_0, F)$ reconnaissant L et la propriété de détermination fera le reste.

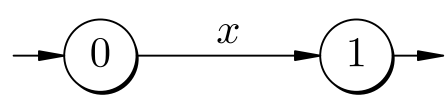
Cette construction se fait par induction sur la définition des langages réguliers (expressions régulières).

On vérifie sans difficulté que ces constructions sont correctes, c'est–à–dire que l'on a bien $\mathcal{L}(\mathbf{A}) = L$ dans chaque cas.

1) $L = \emptyset$.

Il suffit que $F = \emptyset$ pour que $\mathcal{L}(\mathbf{A}) = \emptyset$, par exemple .

2) $L = x$ pour $x \in \mathcal{A}$.

$\mathbf{A}$ a comme graphe de transition 

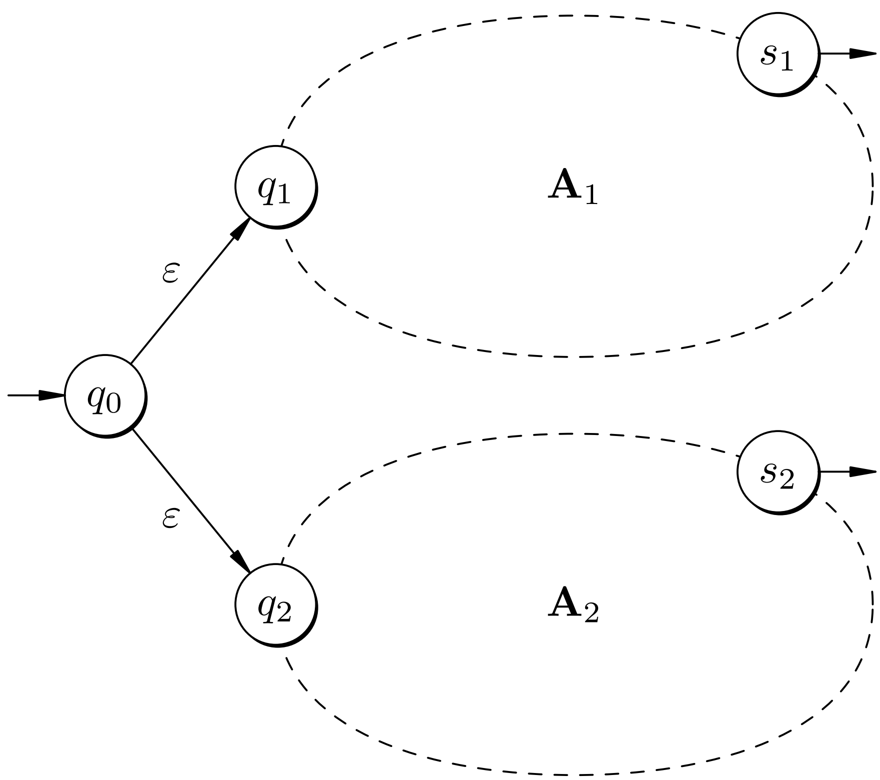
3) $L = L_1 + L_2$ où

L_1 est reconnu par $\mathbf{A}_1 = (Q_1, \mathcal{A}, \delta_1, q_1, F_1)$
L_2 est reconnu par $\mathbf{A}_2 = (Q_2, \mathcal{A}, \delta_2, q_2, F_2)$.

On suppose que $Q_1 \cap Q_2 = \emptyset$.
Par ailleurs, on ne suppose rien d'autre sur $\mathbf{A}_1$ et $\mathbf{A}_2$.

$\mathbf{A}$ est défini par :

- $Q = q_0 + Q_1 + Q_2$ où q_0 est un nouvel état,
- q_0 est le nouvel état qui vient d'être introduit,
- δ prolonge δ_1 et δ_2 par $\delta(q_0, \varepsilon) = q_1 + q_2$,
- $F = F_1 + F_2$.



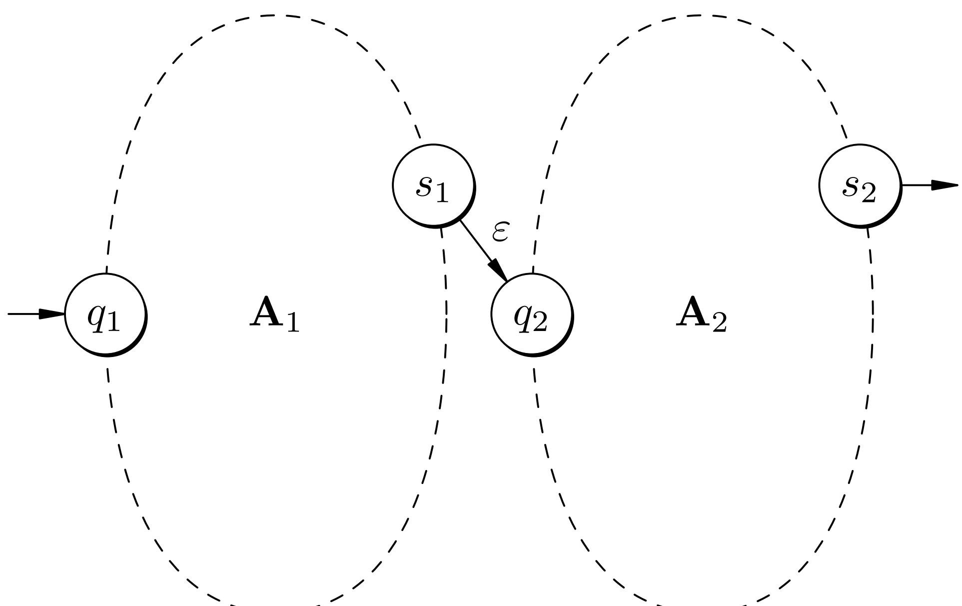
4) $L = L_1L_2$ où

L_1 est reconnu par $\mathbf{A}_1 = (Q_1, \mathcal{A}, \delta_1, q_1, F_1)$
L_2 est reconnu par $\mathbf{A}_2 = (Q_2, \mathcal{A}, \delta_2, q_2, F_2)$.

On suppose que $Q_1 \cap Q_2 = \emptyset$.
Par ailleurs, on ne suppose rien d'autre sur $\mathbf{A}_1$ et $\mathbf{A}_2$.

$\mathbf{A}$ est défini par :

- $Q = Q_1 + Q_2$,
- $q_0 = q_1$,
- δ prolonge δ_1 et δ_2 par
  $$\delta(s_1, \varepsilon) = \delta_1(s_1, \varepsilon) + q_2 \text{ pour tout } s_1 \in F_1,$$
- $F = F_2$.

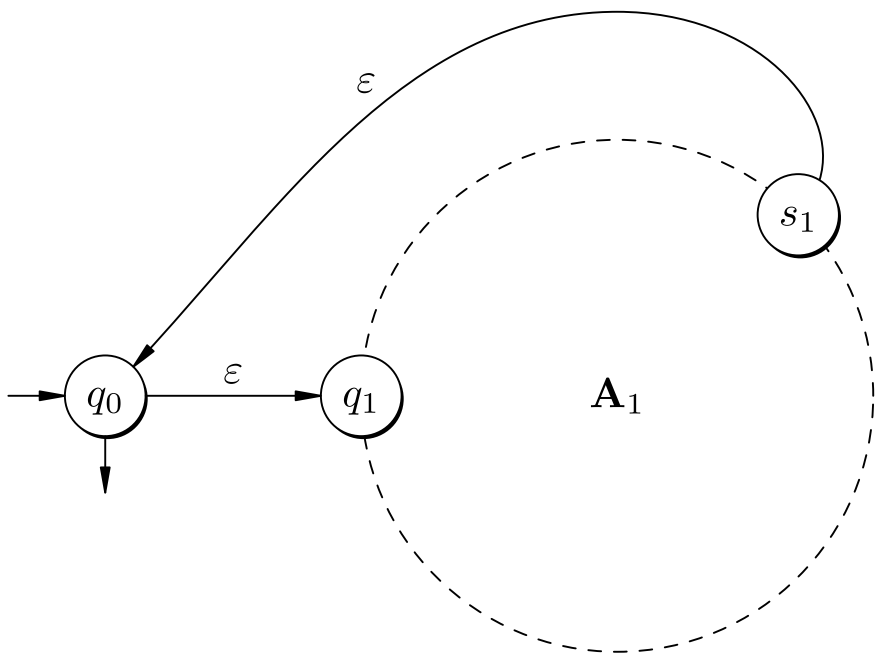


Les ε–transitions jouent un rôle essentiel dans cette construction : lorsqu'il a pénétré dans $\mathbf{A}_2$ un chemin ne doit pas pouvoir entrer à nouveau dans $\mathbf{A}_1$!

5) $L = L_1^*$ où L_1 est reconnu par $\mathbf{A}_1 = (Q_1, \mathcal{A}, \delta_1, q_1, F_1)$.
Par ailleurs, on ne suppose rien d'autre sur $\mathbf{A}_1$.

$\mathbf{A}$ est défini par :

- $Q = q_0 + Q_1$ où q_0 est un nouvel état,
- q_0 est le nouvel état qui vient d'être introduit,
- δ prolonge δ_1 par
  $\delta(q_0, \varepsilon) = q_1$
  $\delta(s_1, \varepsilon) = \delta_1(s_1, \varepsilon) + q_0$ pour tout $s_1 \in F_1$,
- $F = q_0$.



Lorsque q_1 n'est pas une sortie de $\mathbf{A}_1$, on ne peut en général pas éviter d'introduire une nouvelle entrée : si on faisait jouer à q_1 le rôle que tient q_0, on risquerait d'utiliser une éventuelle boucle passant par q_1 ...

5.5 – ε–AF et systèmes d'équations linéaires.

Le système d'équations linéaires, analogue à celui qui nous a servi à faire la preuve du fait que (a) implique (b) dans le théorème de Kleene, peut encore s'écrire dans le cas plus général des ε–AF.

Soit $\mathbf{A} = (Q, \mathcal{A}, \delta, I, F)$ un ε–AF.
Pour tout $q \in Q$ on considère l'ensemble $Rec(q)$ des mots reconnus par l'ε–AF $\mathbf{A}(q) = (Q, \mathcal{A}, \delta, q, F)$.

En étendant Rec aux ensembles d'états, on obtient :

$$Rec(q) = \begin{cases} \displaystyle\sum_{x \in \varepsilon + \mathcal{A}} xRec(\delta(q,x)) + \varepsilon & \text{si } q \in F, \\ \displaystyle\sum_{x \in \varepsilon + \mathcal{A}} xRec((\delta(q,x)) & \text{sinon.} \end{cases}$$

- Lorsque l'on a des ε–transitions, ce système ne vérifie pas la condition d'unicité : nous admettrons ici que les $Rec(q)$ constituent la plus petite solution de ce système.
- Il est clair que $\mathcal{L}(\mathbf{A}) = Rec(I)$.

Exemple 3 (suite).

La table de l'ε–AF de l'exemple 3 est :

e/s	q	$\delta(q,\varepsilon)$	$\delta(q,a)$	$\delta(q,b)$
$\leftrightarrow$	0	1	3	$\emptyset$
	1	$\emptyset$	$1+2$	3
	2	3	4	$\emptyset$
	3	$\emptyset$	$\emptyset$	4
	4	0	$\emptyset$	$\emptyset$

Le système correspondant est le suivant :

$$\begin{aligned}
X_0 &= X_1 + aX_3 + \varepsilon \\
X_1 &= a(X_1 + X_2) + bX_3 \\
X_2 &= X_3 + aX_4 \\
X_3 &= bX_4 \\
X_4 &= X_0
\end{aligned}$$

Sa plus petite solution se calcule par la méthode de Gauss exposée dans le premier chapitre.

6 – Constructions sur les AF.

6.1 – Rôle des sorties d'un AFDC.

Considérons un AFDC "sans sortie" $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0)$; pour tout $F \subseteq Q$, notons $\mathcal{L}(\mathbf{A}, F)$ le langage reconnu par l'AFDC $(Q, \mathcal{A}, \bullet, q_0, F)$:

Propriétés.

1) $\mathcal{L}(\mathbf{A}, \emptyset) = \emptyset$,
2) $\mathcal{L}(\mathbf{A}, Q) = \mathcal{A}^*$,

et, pour tout $F \subseteq Q$ et tout $G \subseteq Q$:

3) $\mathcal{L}(\mathbf{A}, F + G) = \mathcal{L}(\mathbf{A}, F) + \mathcal{L}(\mathbf{A}, G)$,
4) $\mathcal{L}(\mathbf{A}, F \cap G) = \mathcal{L}(\mathbf{A}, F) \cap \mathcal{L}(\mathbf{A}, G)$,
5) $\mathcal{L}(\mathbf{A}, \overline{F}) = \overline{\mathcal{L}(\mathbf{A}, F)}$,

3) est un application directe des définitions ; en effet, pour tout $u \in \mathcal{A}^*$, on a successivement :

$$\begin{aligned} u \in \mathcal{L}(\mathbf{A}, F + G) \text{ ssi } & \quad q_0 \bullet u \in F + G \\ \text{ssi } & \quad q_0 \bullet u \in F \text{ ou } q_0 \bullet u \in G \\ \text{ssi } & \quad u \in \mathcal{L}(\mathbf{A}, F) \text{ ou } u \in \mathcal{L}(\mathbf{A}, G) \\ \text{ssi } & \quad u \in \mathcal{L}(\mathbf{A}, F) + \mathcal{L}(\mathbf{A}, G) \end{aligned}$$

et de même pour 4) en remplaçant les disjonctions par des conjonctions. En remarquant que, pour $F \subseteq Q$ et tout $F' \subseteq Q$, la propriété $F' = \overline{F}$ équivaut à la conjonction "$F + F' = Q$ et $F \cap F' = \emptyset$" et en faisant une remarque analogue pour les parties de $\mathcal{A}^*$, on déduit la propriété 5) des quatre précédentes.

En appliquant le théorème de Kleene, on peut en déduire deux importantes propriétés des langages réguliers :

Propriétés des langages réguliers

Soient L et M deux langages réguliers sur $\mathcal{A}$, alors :

- le complémentaire $\overline{L}$ de L est un langage régulier,
- l'intersection $L \cap M$ est un langage régulier.

Il suffit d'appliquer la propriété 5) qui précède à un AFDC reconnaissant L pour vérifier la première propriété.

Pour la seconde : si L et M sont réguliers, leurs complémentaires $\overline{L}$ et $\overline{M}$ le sont aussi par la première propriété, leur somme $N = \overline{L} + \overline{M}$ est donc un langage régulier : le complémentaire de N est donc régulier or, on sait que $\overline{N} = L \cap M$.

La construction suivante, qui est intéressante pour elle–même, conduit aussi à ce résultat.

6.2 – Produit d'AFDC.

Le produit $\mathbf{A} \times \mathbf{B} = (Q, \mathcal{A}, \circ, q_0, F)$ des AFDC

$$\mathbf{A} = (Q^{\mathbf{A}}, \mathcal{A}, \underset{\mathbf{A}}{\bullet}, q_0^{\mathbf{A}}, F^{\mathbf{A}})$$

et

$$\mathbf{B} = (Q^{\mathbf{B}}, \mathcal{A}, \underset{\mathbf{B}}{\bullet}, q_0^{\mathbf{B}}, F^{\mathbf{B}})$$

est l'AFDC défini de la façon suivante :

Etats. $Q = Q^{\mathbf{A}} \times Q^{\mathbf{B}}$

Action. $(p, q) \circ x = (p \underset{\mathbf{A}}{\bullet} x, q \underset{\mathbf{B}}{\bullet} x)$ pour tout $p \in Q^{\mathbf{A}}$ et tout $q \in Q^{\mathbf{B}}$,

Entrée. $q_0 = (q_0^{\mathbf{A}}, q_0^{\mathbf{B}})$,

Sorties. $F = F^{\mathbf{A}} \times F^{\mathbf{B}}$.

La propriété essentielle de cette opération est

$$\mathcal{L}(\mathbf{A} \times \mathbf{B}) = \mathcal{L}(\mathbf{A}) \cap \mathcal{L}(\mathbf{B}).$$

En effet, pour chaque $u \in \mathcal{A}^*$:

$$\begin{aligned}
u \in \mathcal{L}(\mathbf{A} \times \mathbf{B}) \text{ ssi } \quad & q_0 \circ u \in F \\
\text{ssi } \quad & (q_0^{\mathbf{A}}, q_0^{\mathbf{B}}) \circ u \in F^{\mathbf{A}} \times F^{\mathbf{B}} \\
\text{ssi } \quad & (q_0^{\mathbf{A}} \underset{\mathbf{A}}{\bullet} u, q_0^{\mathbf{B}} \underset{\mathbf{B}}{\bullet} u) \in F^{\mathbf{A}} \times F^{\mathbf{B}} \\
\text{ssi } \quad & q_0^{\mathbf{A}} \underset{\mathbf{A}}{\bullet} u \in F^{\mathbf{A}} \text{ et } q_0^{\mathbf{B}} \underset{\mathbf{B}}{\bullet} u \in F^{\mathbf{B}} \\
\text{ssi } \quad & u \in \mathcal{L}(\mathbf{A}) \text{ et } u \in \mathcal{L}(\mathbf{B}) \\
\text{ssi } \quad & u \in \mathcal{L}(\mathbf{A}) \cap \mathcal{L}(\mathbf{B})
\end{aligned}$$

7 – Minimisation des AFDC.

Parmi les AFDC reconnaissant un langage régulier donné L, il en existe dont le nombre d'états $|Q|$ est le plus petit possible (un ensemble d'entiers, non vide, a un plus petit élément) : en fait, il n'en existe qu'un seul, au renommage près des états, que l'on appelle l'AFDC minimal de L.

Nous décrivons la construction de l'AFDC minimal

$$Min(\mathbf{A}) = (Q', \mathcal{A}, \circ, q'_0, F')$$

équivalent à un AFDC $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ donné. Elle s'applique lorsque **tous les états de A sont accessibles** (l'élimination des états inaccessibles est déjà une étape vers la minimisation !).

Un peu de musique : partition d'un ensemble.

• Une *partition* Π d'un ensemble $Q \neq \emptyset$ est un ensemble de parties $\Pi \subseteq \mathcal{P}(Q)$ qui vérifie :

– pour toute $S \in \Pi$: $S \neq \emptyset$
– pour toute $S \in \Pi$ et toute $T \in \Pi$: $S = T$ ou $S \cap T = \emptyset$
– pour tout $q \in Q$, il existe $S \in \Pi$ telle que $q \in S$

c'est–à–dire le résultat d'un découpage de Q.

• Soit Π est une partition de Q alors, tout $q \in Q$ appartient à un élément de Π et à un seul que l'on notera $[q]$. Pour toute $S \in \Pi$ et tout $q \in Q$ on a donc la propriété :

$$[q] = S \text{ ssi } q \in S.$$

7.1 – Construction de Q'.

On utilise l'action inverse $s \bullet x^{-1} \subseteq Q$ de $x \in \mathcal{A}$ sur $s \in Q$:

$$q \in s \bullet x^{-1} \text{ ssi } q \bullet x = s.$$

Son extension à $S \subseteq Q$ est définie par $S \bullet x^{-1} = \sum_{s \in S} s \bullet x^{-1}$

et vérifie donc

$$q \in S \bullet x^{-1} \text{ ssi } q \bullet x \in S$$

Calcul d'une partition de Q

- *La valeur initiale de* Π *est*

$$\Pi = \begin{cases} \{Q\} & \textit{si } F = Q \textit{ ou } F = \emptyset, \\ \{F, Q - F\} & \textit{sinon.} \end{cases}$$

- *tant qu'il existe* $S \in \Pi$, $T \in \Pi$ *et* $x \in \mathcal{A}$ *tels que*

$$S_1 = S \cap (T \bullet x^{-1}) \neq \emptyset \textit{ et } S_2 = S - (T \bullet x^{-1}) \neq \emptyset$$

on raffine Π *en remplaçant* S *par les deux classes* S_1 *et* S_2.

7.2 – Construction de l'AFDC $Min(\mathbf{A})$.

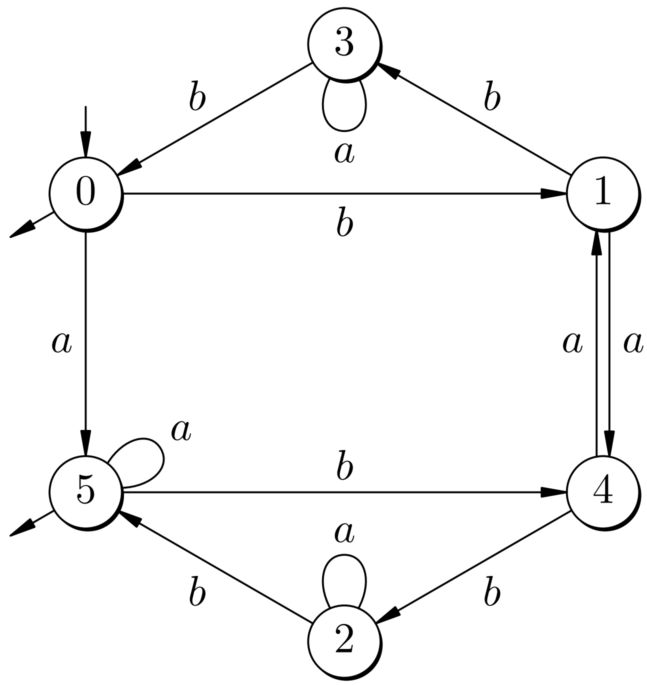
Soit $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ un AFDC **dont tous les états sont accessibles** :

$Min(\mathbf{A})$

$Min(\mathbf{A})$ est l'AFDC $(Q', \mathcal{A}, \circ, q'_0, F')$ suivant :

- *Etats.* Q' est la partition Π calculée par l'algorithme.
- *Action.* $[q] \circ x = [q \bullet x]$.
- *Entrée.* $q'_0 = [q_0]$.
- *Sorties.* $[q] \in F'$ ssi $q \in F$ (ssi $[q] \subseteq F$).

Exemple.



La table de l'action réciproque de cet AFDC est :

e/s	q	$q \bullet a^{-1}$	$q \bullet b^{-1}$
$\leftrightarrow$	0	$\emptyset$	3
	1	4	0
	2	2	4
	3	3	1
	4	1	5
$\leftarrow$	5	$0+5$	2

La partition initiale est : $\Pi = \{0+5, 1+2+3+4\}$.
$(0+5) \bullet a^{-1} = 0 \bullet a^{-1} + 5 \bullet a^{-1} = \emptyset + (0+5) = 0+5$
qui ne permet pas de raffiner Π.
$(0+5) \bullet b^{-1} = 0 \bullet b^{-1} + 5 \bullet b^{-1} = 3+2 = 2+3$ permet de raffiner Π en $\Pi = \{0+5, 1+4, 2+3\}$.
Enfin :

$$\begin{aligned}
(1+4) \bullet a^{-1} &= 1+4 \\
(1+4) \bullet b^{-1} &= 0+5 \\
(2+3) \bullet a^{-1} &= 2+3 \\
(2+3) \bullet b^{-1} &= 1+4
\end{aligned}$$

Ces valeurs ne permettent plus de raffiner la partition, donc :

$$Q' = \Pi = \{0+5, 1+4, 2+3\}.$$

e/s	S	$S \circ a$	$S \circ b$
$\leftrightarrow$	$A = 0+5$	A	B
	$B = 1+4$	B	C
	$C = 2+3$	C	A