

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

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

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.

 l 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 $\mathcal{A} = a + b$:
 - $\emptyset, \varepsilon, a, b, \mathcal{A} = a + b$ 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

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

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 $\bullet : Q \times \mathcal{A} \rightarrow Q$ s’étend de façon unique en une application

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

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 $\bullet : Q \times \mathcal{A} \rightarrow Q$ 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 $v \in \mathcal{A}^*$.

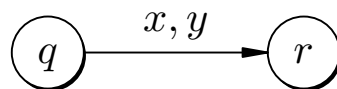
Il y a deux représentations pratiques possibles d'une action :

- 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 :

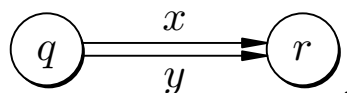
- un nœud (q) pour tout $q \in Q$;
- une arête $(q) \xrightarrow{x} (r)$ pour tout $q \in Q$, tout $x \in \mathcal{A}$ et $r = q \bullet x$.

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 :



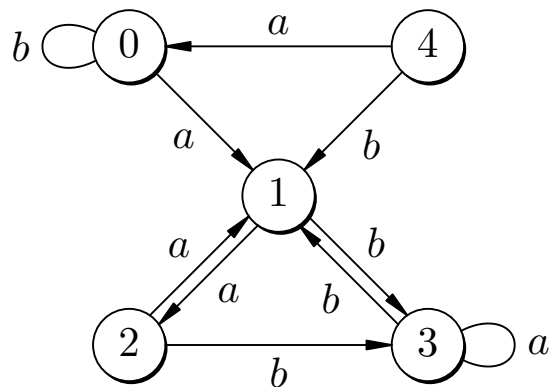
représente le couple d'arêtes



Exemple 1.

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

q	$q \cdot a$	$q \cdot b$
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	$q \cdot a$	$q \cdot b$
\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 :

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

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

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

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

- *Etats.* $Q' = \mathcal{Acc}$ (tous les états accessibles dans \mathbf{A})
- *Entrée.* q_0 (état initial de \mathbf{A})
- *Action.* $\bullet : Q' \times \mathcal{A} \rightarrow Q'$ (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).

$$\mathcal{U}_0 = 0$$

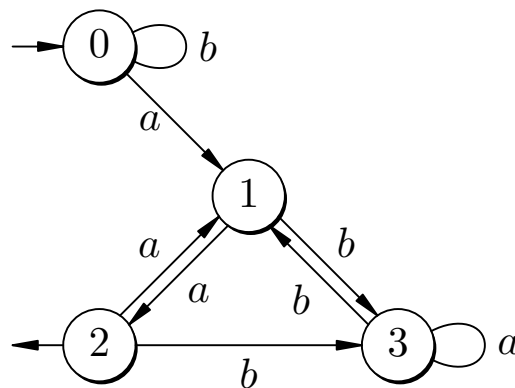
$$\mathcal{U}_1 = 0 + 0 \cdot (a + b) = 0 + 0 \cdot a + 0 \cdot b = 0 + 1 + 0 = 0 + 1$$

$$\mathcal{U}_2 = 0 + 1 + (0 + 1) \cdot (a + b) = 0 + 1 + 2 + 3$$

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

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

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



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

Calcul de Acc

```

Acc := q0
tant qu'il existe q ∈ Acc non marqué faire
    sélectionner q ∈ Acc non marqué
    pour x parcourant  $\mathcal{A}$  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 $Acc := Acc + q \cdot x$ est exécutée, Acc n'est pas modifié si $q \cdot x \in Acc$.

Remarque.

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

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 :

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

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 $X_q = \text{Rec}(q)$ et dont les coefficients sont réguliers (ε et des sommes finies d'éléments de \mathcal{A}); de plus la condition d'unicité est vérifiée :

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

Exemple 1 (suite).

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

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

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 AFDC 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 AFDC !

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

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

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

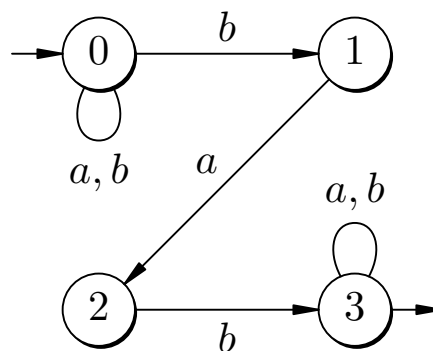
$q \bullet x$ peut donc :

- 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 AFDC.

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	$q \cdot a$	$q \cdot b$
\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,
 - $\bullet : Q \times \mathcal{A} \rightarrow \mathcal{P}(Q)$ 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 $\bullet : Q \times \mathcal{A} \rightarrow \mathcal{P}(Q)$ 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 \textcircled{q} pour tout $q \in Q$;
- une arête $\textcircled{q} \xrightarrow{x} \textcircled{r}$ 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 $r \in Q$:

$$(q, xu) \vdash_{\mathbf{A}}^1 (r, u) \text{ ssi } r \in q \bullet x$$

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

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

$$0 : (q, u) \vdash_{\mathbf{A}}^0 (q, u) \quad (\text{pas de transition})$$

$$i \mapsto i + 1 : (q, u) \vdash_{\mathbf{A}}^i (s, w) \vdash_{\mathbf{A}}^1 (r, v) \quad (\text{une transition de plus})$$

- On définit la relation $\vdash_{\mathbf{A}}^*$ entre configurations par :

$$(q, u) \vdash_{\mathbf{A}}^* (r, v) \text{ ssi il existe une dérivation } (q, u) \vdash_{\mathbf{A}}^i (r, v).$$

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

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

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

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

Dans un AFDC

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

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 $(q, u) \vdash^i (q \bullet u, \varepsilon)$, 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

$$(0, abab) \vdash^1 (0 \bullet a, bab) = (1, bab)$$

$$\vdash^1 (1 \bullet b, ab) = (3, ab)$$

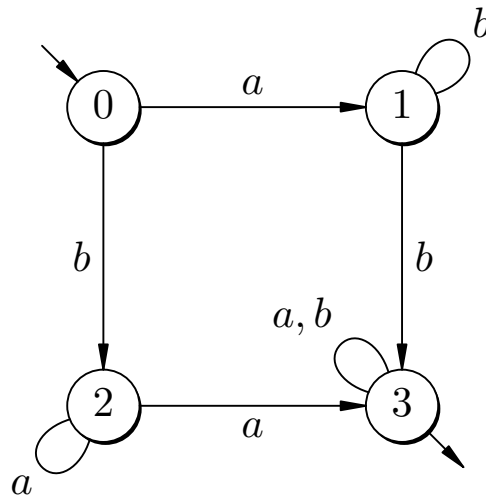
$$\vdash^1 (3 \bullet a, b) = (3, b)$$

$$\vdash^1 (3 \bullet b, \varepsilon) = (1, \varepsilon)$$

Exemple 2.

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

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



Dans cet AF, on a les dérivations

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

mais $(0, abba) \stackrel{1}{\vdash} (1, bba) \stackrel{1}{\vdash} (1, ba) \stackrel{1}{\vdash} (1, a)$ tombe dans une impasse puisque $1 \cdot a = \emptyset$.

4.2 – Détermination.

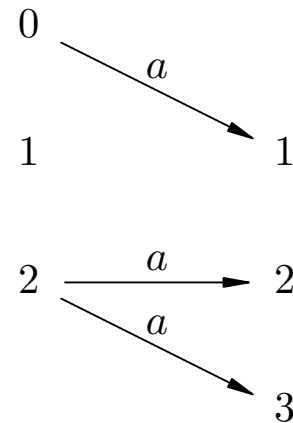
On peut simuler les tentatives pour construire un chemin de façon systématique en considérant l'extension

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

aux ensembles d'états de l'application $\bullet : Q \times \mathcal{A} \rightarrow \mathcal{P}(Q)$ 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 $r \in Q$:

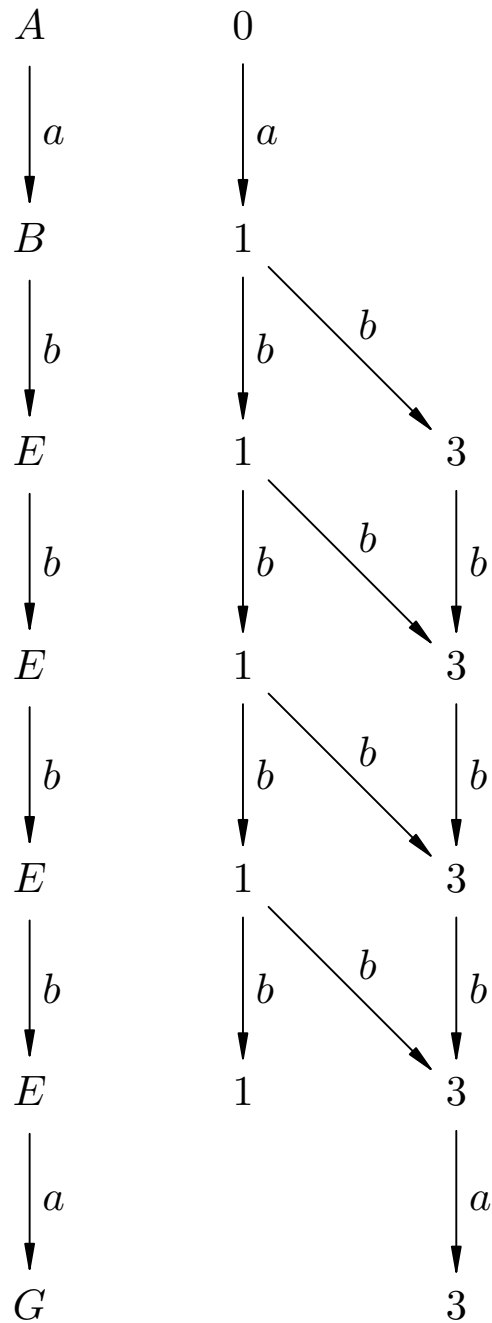
$$r \in S \bullet u \text{ ssi il existe } s \in S \text{ tel que } (s, u) \vdash^* (r, \varepsilon),$$
- 2) $u \in \mathcal{L}(\mathbf{A})$ ssi $(I \bullet u) \cap F \neq \emptyset$.

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.* $\bullet : Q' \times \mathcal{A} \rightarrow Q'$, l'extension de \bullet
aux ensembles d'états,
 - *Entrée.* $q'_0 = I$,
 - *Sorties.* $S \in F'$ ssi $S \cap F \neq \emptyset$.
-

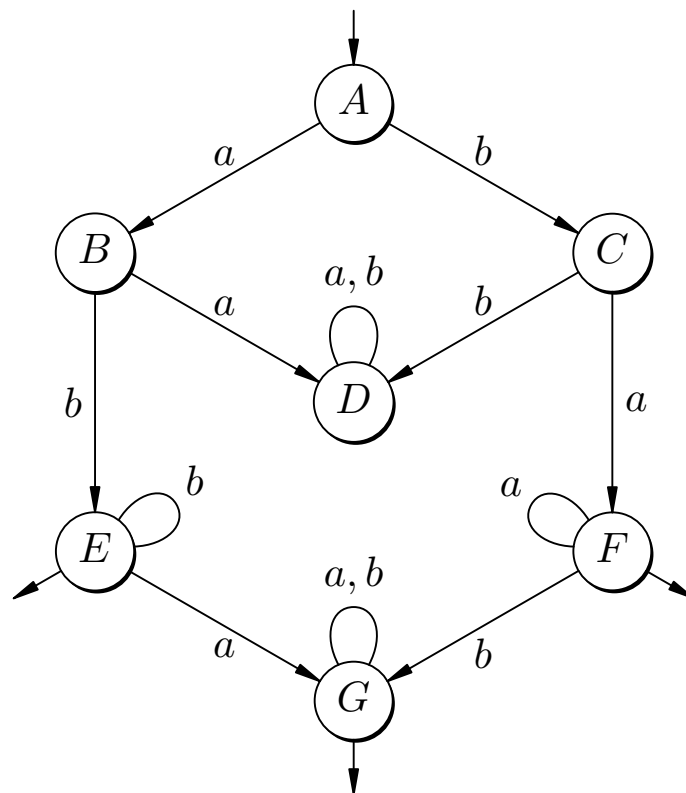
Le fait que $DC(\mathbf{A})$ est équivalent à \mathbf{A} est exprimé par les propriétés ci-dessus.

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	$S \cdot a$	$S \cdot b$
\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 $(q, u) \vdash^* (r, \varepsilon)$.

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
 $(q \bullet x) \cap \mathcal{P}rod \neq \emptyset$ 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.* $\circ : Q' \times \mathcal{A} \rightarrow Q'$, ($q \circ x = (q \bullet x) \cap \mathcal{P}rod$)
- *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 $Prod$ est la limite de la suite croissante stationnaire \mathcal{U}_i définie par la récurrence suivante :

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

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

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

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

Attention.

Lorsque \mathbf{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

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

- 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 \textcircled{q} pour tout $q \in Q$;
- une arête $\textcircled{q} \xrightarrow{x} \textcircled{r}$ 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 $u \in \mathcal{A}^*$:

$$(q, xu) \vdash_{\mathbf{A}}^1 (r, u) \text{ ssi } r \in \delta(q, x)$$

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 $(q, u) \vdash_{\mathbf{A}}^i (r, v)$ est un enchaînement de i transitions successives.

- Une ε -*dérivation* est une dérivation de la forme

$$(q, u) \vdash_{\mathbf{A}}^i (r, u)$$

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

- On définit la relation $\vdash_{\mathbf{A}}^*$ entre configurations par :

$$(q, u) \vdash_{\mathbf{A}}^* (r, v) \text{ ssi il existe une dérivation } (q, u) \vdash_{\mathbf{A}}^i (r, v).$$

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

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

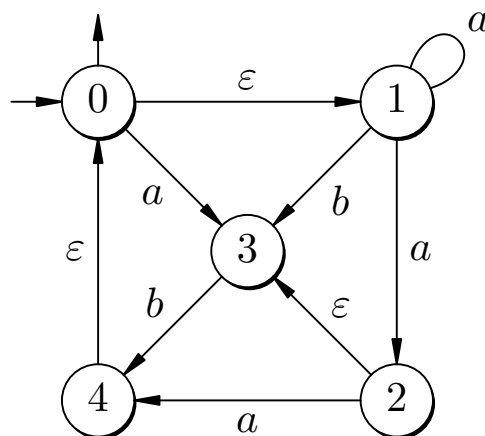
Deux ε -AF \mathbf{A} et \mathbf{A}' sont *équivalents* ssi $\mathcal{L}(\mathbf{A}) = \mathcal{L}(\mathbf{A}')$.

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 \mathbf{A} .

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

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

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

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

L'existence de la dérivation triviale $(q, \varepsilon) \vdash^0 (q, \varepsilon)$ 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.* $\bullet : Q' \times \mathcal{A} \rightarrow Q'$ 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 :

$$cl(1) = 1 \subseteq B$$

$$cl(2) = 2 + 3 \subseteq B$$

$$cl(3) = 3 \subseteq B$$

donc $cl(B) \subseteq B$.

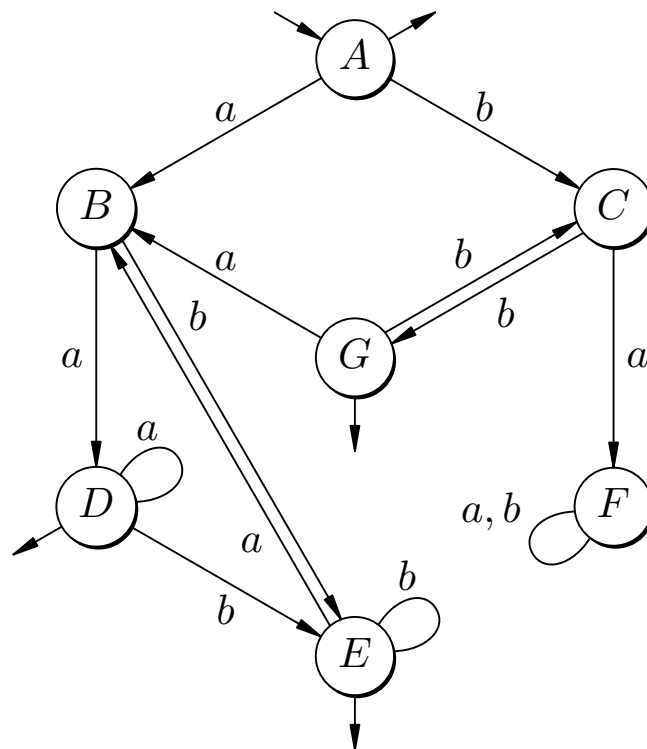
- Voici les actions sur B :

$$B \cdot a = cl(1 + 2) + cl(4) + \emptyset = 0 + 1 + 2 + 3 + 4$$

$$B \cdot b = cl(3) + \emptyset + cl(4) = 0 + 1 + 3 + 4.$$

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	$S \cdot a$	$S \cdot b$
\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

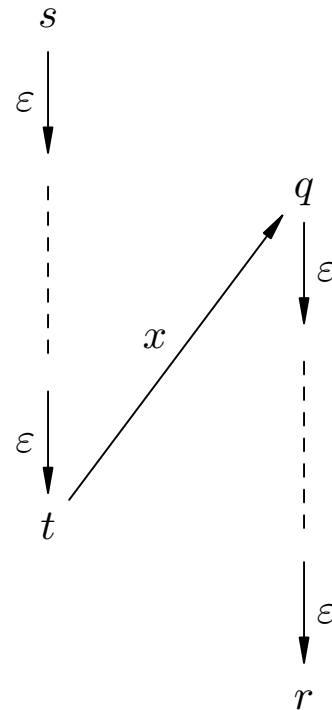
il existe $s \in S$ tel que
 $(s, u) \vdash^* (r, \varepsilon)$,

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

ssi

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

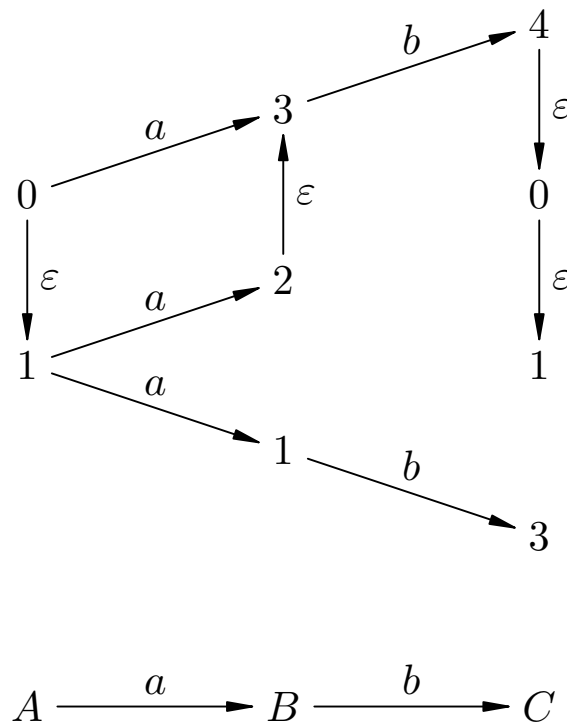
La figure ci-contre illustre 1)
pour $u = x \in \mathcal{A}$.



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

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 $\rightarrow \textcircled{0}$.

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

\mathbf{A} a comme graphe de transition $\rightarrow \textcircled{0} \xrightarrow{x} \textcircled{1} \rightarrow$

3) $L = L_1 + L_2$ où

L_1 est reconnu par $\mathbf{A}_1 = (Q_1, \mathcal{A}, \delta_1, q_1, F_1)$

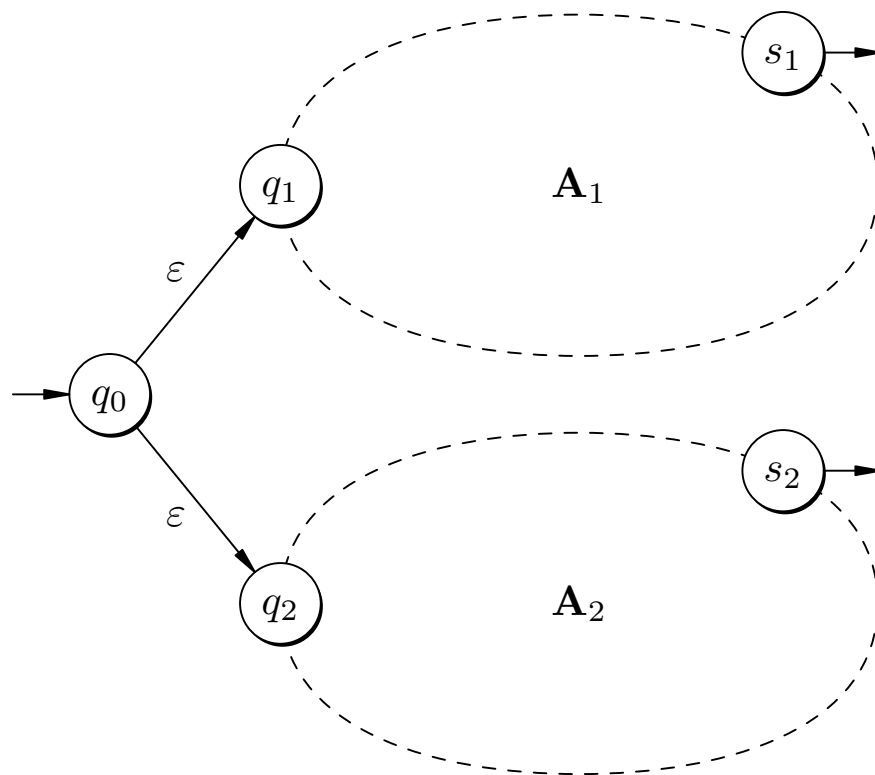
L_2 est reconnu par $\mathbf{A}_2 = (Q_2, \mathcal{A}, \delta_2, q_2, F_2)$.

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 $\mathbf{A}_2 = (Q_2, \mathcal{A}, \delta_2, q_2, F_2)$.

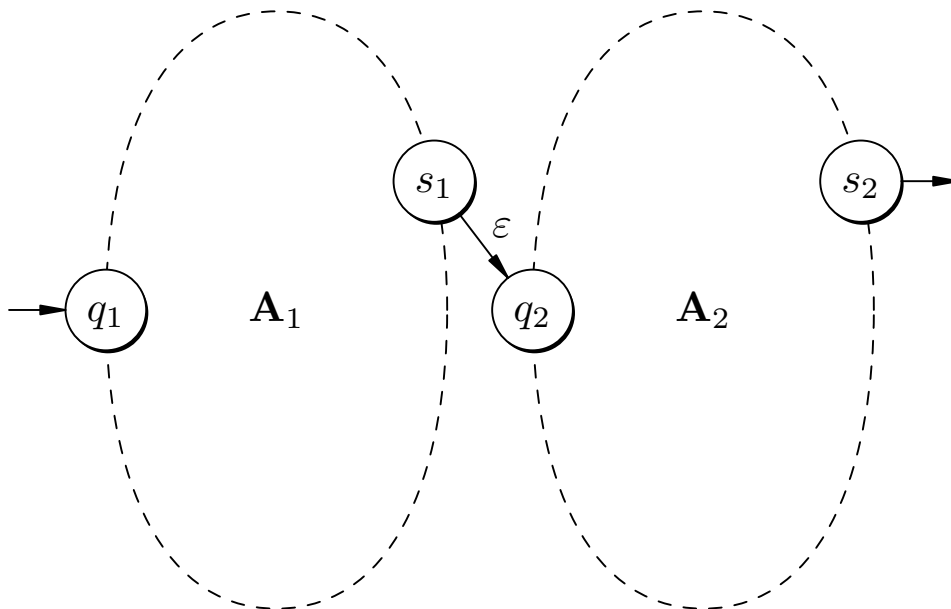
On suppose que $Q_1 \cap Q_2 = \emptyset$.

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

$$\delta(s_1, \varepsilon) = \delta_1(s_1, \varepsilon) + q_2 \text{ pour tout } s_1 \in F_1,$$
- $F = F_2$.



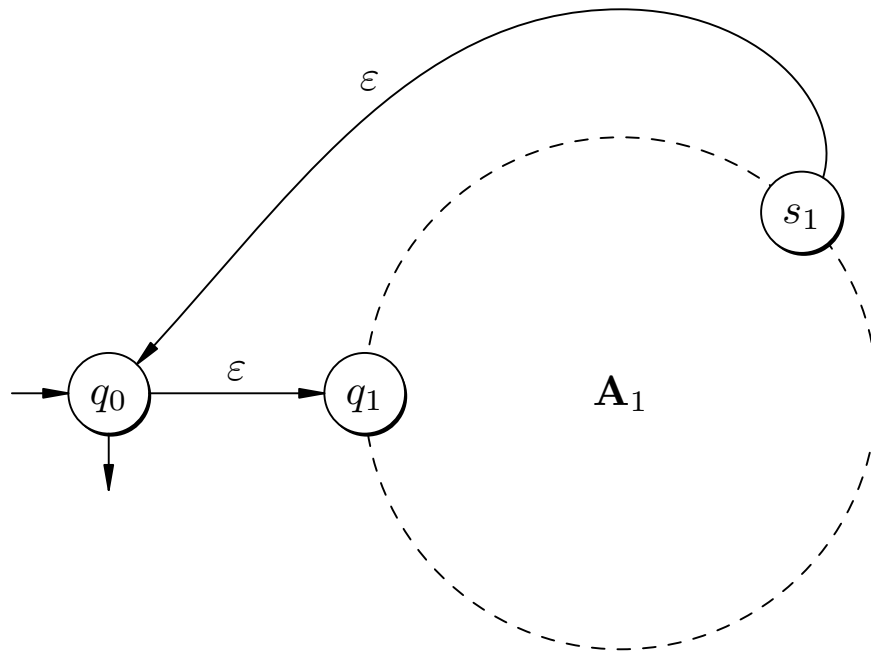
Les ε -transitions jouent un rôle essentiel dans cette construction : lorsqu'il a pénétré dans \mathbf{A}_2 un chemin ne doit pas pouvoir entrer à nouveau dans \mathbf{A}_1 !


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 $s_1 \in F_1$,
- $F = q_0$.




 orsque 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 :

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

- 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 :

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

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

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

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

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

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

équivalent à un AFDC $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ donné. Elle s'applique lorsque **tous les états de \mathbf{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

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

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

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

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

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

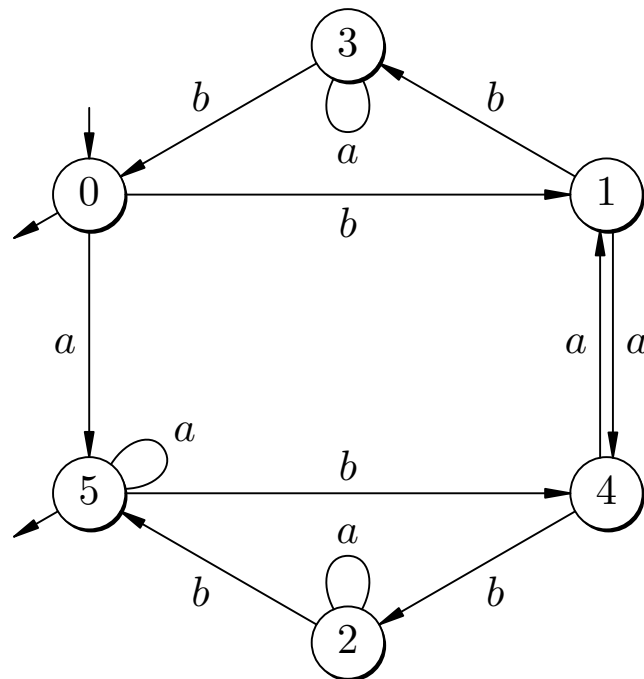
Soit $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ un AFDC dont tous les états sont accessibles :

$Min(\mathbf{A})$

$Min(\mathbf{A})$ 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	$q \cdot a^{-1}$	$q \cdot b^{-1}$
\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 : $\Pi = \{0 + 5, 1 + 2 + 3 + 4\}$.

$$(0 + 5) \cdot a^{-1} = 0 \cdot a^{-1} + 5 \cdot a^{-1} = \emptyset + (0 + 5) = 0 + 5$$

qui ne permet pas de raffiner Π .

$(0 + 5) \cdot b^{-1} = 0 \cdot b^{-1} + 5 \cdot b^{-1} = 3 + 2 = 2 + 3$ permet de raffiner Π en $\Pi = \{0 + 5, 1 + 4, 2 + 3\}$.

Enfin :

$$(1 + 4) \cdot a^{-1} = 1 + 4$$

$$(1 + 4) \cdot b^{-1} = 0 + 5$$

$$(2 + 3) \cdot a^{-1} = 2 + 3$$

$$(2 + 3) \cdot b^{-1} = 1 + 4$$

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

