CH.2 Minimisation

- 2.1 L'automate minimal
- 2.2 L'algorithme de minimisation

2.1 L'automate minimal

Le langage L définit sur Σ^* la relation d'équivalence R_L :

$x\ R_L\ y$ ssi $(\forall z, xz \in L \Leftrightarrow yz \in L)$.

L'AFD M définit sur Σ^* la relation d'équivalence R_M :

$x\ R_M\ y$ ssi $\delta(q_0, x) = \delta(q_0, y)$.

Une relation R est invariante à droite si

$x\ R\ y \Rightarrow (\forall z, xz\ R\ yz)$.

L'indice de R est le nombre de classes d'équivalence de R.

Lemme : Les relations R_L et R_M sont invariantes à droite.

Démonstration : facile.

Lemme : La relation R_M est d'indice fini.

Démonstration : les classes d'équivalence de R_M sont en bijection avec les états de M.

Théorème : Equivalence entre
1.— L est accepté par un AFD ;
2.— L est réunion de classes d'équivalence d'une relation invariante à droite d'indice fini ;
3.— La relation R_L est d'indice fini.

Démonstration :
1. $\Rightarrow$ 2.
Soit M un AFD acceptant L. Alors R_M est invariante à droite et L est réunion des classes de R_M correspondant aux états terminaux de M.
2. $\Rightarrow$ 3.
Supposons que L est réunion de classes de R d'indice fini.
Montrons que x R y $\Rightarrow$ x R_L y.
Si x R y, on a xz R yz (invariance à droite). Les mots xz et yz sont donc dans la même classe de R. Soit cette classe est une des classes qui constitue L, auquel cas xz et yz sont tous deux dans L, soit ce n'est pas le cas et ni xz ni yz ne sont dans L. Dans tous les cas, et quel que soit z, on a xz $\in$ L $\Leftrightarrow$ yz $\in$ L. Donc x R_L y.
On en déduit qu'une classe de R est incluse dans une classe de R_L.
Chaque classe de R_L est donc réunion de classes de R.
R_L est donc d'indice fini.

3. $\Rightarrow$ 1.
R_L est d'indice fini.
Notons c_0 la classe de ε et $c_1,\ldots,c_n$ les autres classes.
Définissons $\delta(c_i, a)$. Pour cela soit x un mot dans c_i. Posons $\delta(c_i, a)$ = la classe de xa. Il faut vérifier que cette classe ne dépend pas du choix de x. Ceci résulte de l'invariance à droite de R_L. La fonction δ est donc définie. Disons maintenant que c_k est terminal si tous les mots de c_k sont dans L. (Si un élément de c_k est dans L, alors tous le sont, car R_L est invariante à droite par ε.)
On a ainsi défini un automate fini M. Montrons que M accepte L.
On établit par récurrence sur |x| que $\delta(c_0, x)$ = la classe de x, disons c_k.
On a donc $x \in L(M) \Leftrightarrow \delta(c_0, x) = c_k$ terminal $\Leftrightarrow x \in L$.
Ceci achève la démonstration du théorème.

Théorème de Myhill-Nerode : Soit L un langage reconnaissable. Parmi tous les AFD reconnaissant L, il en existe un et un seul (au nom des états près) qui a un nombre minimum d'états. Cet automate est l'automate minimal de L.

Démonstration : Soit M un automate ayant un nombre minimum d'états et M_L l'automate construit à partir de R_L. On a vu que les états de M_L sont les classes de R_L ; d'autre part, les classes de R_L sont réunion de classes de R_M, qui correspondent elles-mêmes aux états de M. On voit que le nombre d'états de M_L est inférieur au nombre d'états de M. Puisque le nombre d'états de M est minimum, il y a égalité. Chaque état de R_L est en bijection avec un état de M. Vérifions que les transitions sont identiques.

Soit c_i dans M_L qui correspond à q_i, dans M. Soit x un mot dans la classe c_i de R_L ; il est donc dans la classe associée à q_i pour R_M.
C'est-à-dire $\delta(q_0, x) = q_i$ dans M.
On a, dans M, $\delta(q_i, a) = \delta(q_0, xa)$ = état associé à la classe de xa pour R_M.
On a de même, dans M_L, $\delta(c_i, a)$ = classe de xa pour R_L.
Puisque la classes de R_M et de R_L coïncident, $\delta(q_i, a)$ et $\delta(c_i, a)$ sont deux états correspondants des automates M et M_L.
Les deux automates sont donc identiques, au numérotage près.

Remarque : On a montré l'existence et l'unicité de l'automate minimal.
La démonstration n'est pas immédiatement convertible en un algorithme.

2.2 L'algorithme de minimisation

Soit M un automate déterministe reconnaissant L.
Supposons que les états de M sont tous accessibles.

Les états de M_L correspondent aux classes de R_L. Celles-ci sont réunion de classes de R_M, qui correspondent aux états de M.

Les états de M_L peuvent donc être vus comme des réunions de classes de R_M, et donc comme "fusion" d'états de M.

Plus précisément, deux classes de R_M correspondant aux états q_i et q_j sont composantes de la même classe de R_L lorsque, étant donnés x et y tels que $\delta(q_0, x) = q_i$ et $\delta(q_0, y) = q_i$,
$\forall z, xz \in L \Leftrightarrow yz \in L$, soit de façon équivalente,
$\forall z, \delta(q_i, z) \in F \Leftrightarrow \delta(q_j, z) \in F$.

On définit donc sur les états de M une relation d'équivalence, l'indistingabilité en disant que
$q_i \approx q_j$ ssi $(\forall z, \delta(q_i, z) \in F \Leftrightarrow \delta(q_j, z) \in F)$.

Donc, q_i et q_j sont distingables s'il existe z tel que des deux états $\delta(q_i, z)$ et $\delta(q_j, z)$, l'un est terminal et pas l'autre.

On peut définir la distingabilité de manière récursive :
Les états q_i et q_j sont distingables
- si l'un est terminal et pas l'autre

ou
- s'il existe une lettre a telle que $\delta(q_i, a)$ et $\delta(q_j, a)$ sont distingables.

Les états indistingables sont ceux qui fusionnnent en un unique état de l'automate minimal M_L.

On construit donc les paires d'états dinstingables à partir des états terminaux et non terminaux :

Au départ, les paires {terminal, non terminal} sont marquées D.
Les paires marquées D sont placées dans un tas (ou une file).
On dépile la première paire. S'il existe une lettre a et une paire d'états non encore marquée qui devient distingable par a, on marque cette paire et on l'empile.
On itère jusqu'à ce que la pile soit vide.

Les paires marquées sont exactement les paires distingables.

L'algorithme est celui de Moore.

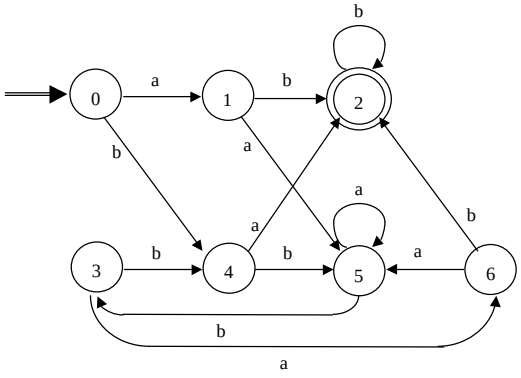
Théorème : La complexité de l'algorithme de Moore est $O(kn^2)$, où k est le cardinal de l'alphabet et n le nombre d'états de M.

Démonstration : Dans la pile, chaque paire d'états est empilée au plus une fois. Pour chaque paire dépilée $\{q_i, q_j\}$, on examine pour chaque lettre a les paires d'états $\{q_r, q_s\}$ tels que
$\delta(q_r, a) = q_i$ et $\delta(q_s, a) = q_j$.
Si f(i, a) est le nombre de transitions par a arrivant en q_i, on examine au maximum $\sum_{a \in \Sigma} f(i,a)\, f(j, a)$ paires d'états par paire dépilée. On examine en tout au maximum :
$\sum_{i<j} \sum_{a \in \Sigma} f(i,a)\, f(j, a) \leq \sum_{i,j} \sum_{a \in \Sigma} f(i,a)\, f(j, a) = \sum_{a \in \Sigma} (\sum_i f(i,a))^2$.
Comme M est déterministe, le nombre de transitions sur a vaut n :
$\sum_i f(i,a) = n$, d'où le résultat.

On peut descendre à une complexité en O(kn log n) en gérant plus astucieusement l'examen des transitions.

Exemple :

0							
1	D						
2	D	D					
3		D	D				
4	D	D	D	D			
5	D	D	D	D	D		
6	D		D	D	D	D	
	0	1	2	3	4	5	6



Automate minimal :