Automates & Calculabilité — Corrigé Série 8 — 8 déc 2003

1. Minimisation des états d'AFD

Pour chaque AFD donné ci-dessous, trouver les classes d'équivalence et dessiner l'automate minimal respectif. (S désigne l'état initial et F désigne un état final)

	a	b
S1	1	4
2	3	1
F3	4	2
F4	3	5
5	4	6
6	6	3
7	2	4
8	3	1

	a	b
SF1	3	5
F2	8	7
3	7	2
4	6	2
5	1	8
6	2	3
7	1	4
8	5	1

Conseil : Minimisez d'abord le premier automate, puis passez au deuxième exercice. Ensuite, si vous avez encore le temps, minimisez le second.

Corrigé :

1. On doit d'abord éliminer les états qui ne sont pas atteignables. Si on examine les transitions de l'automate, on voit que les états 7 et 8 ne sont jamais atteints. On peut donc déjà réduire l'automate en :

	a	b
S1	1	4
2	3	1
F3	4	2
F4	3	5
5	4	6
6	6	3

   On peut maintenant appliquer l'algorithme de minimisation :

1					
-	2				
-	-	3			
-	-	-	4		
-	-	-	-	5	
-	-	-	-	-	6

   On marque tous les couples d'états qui ont un état final et un autre non-final.

1					
-	2				
$\surd$	$\surd$	3			
$\surd$	$\surd$	-	4		
-	-	$\surd$	$\surd$	5	
-	-	$\surd$	$\surd$	-	6

On regarde les transitions possibles depuis les couples qui ne sont pas encore marqués. On met en évidence les états déjà marqués.

input a :	$\{1,2\}$	$\rightarrow$	$\{1,3\}$	$\surd$
input a :	$\{1,5\}$	$\rightarrow$	$\{1,4\}$	$\surd$
input a :	$\{2,6\}$	$\rightarrow$	$\{3,6\}$	$\surd$
input a :	$\{5,6\}$	$\rightarrow$	$\{4,6\}$	$\surd$
input a :	$\{1,6\}$	$\rightarrow$	$\{1,6\}$	
input b :	$\{1,6\}$	$\rightarrow$	$\{4,3\}$	
input a :	$\{3,4\}$	$\rightarrow$	$\{4,3\}$	
input b :	$\{3,4\}$	$\rightarrow$	$\{2,5\}$	
input a :	$\{2,5\}$	$\rightarrow$	$\{3,4\}$	
input b :	$\{2,5\}$	$\rightarrow$	$\{1,6\}$	

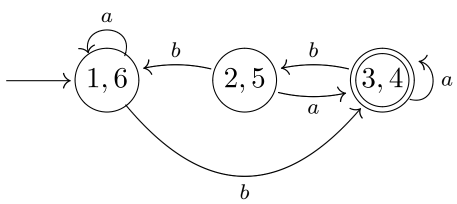
Après la première étape on peut marquer quatre autres états :

1					
$\surd$	2				
$\surd$	$\surd$	3			
$\surd$	$\surd$	-	4		
$\surd$	-	$\surd$	$\surd$	5	
-	$\surd$	$\surd$	$\surd$	$\surd$	6

On examine les transitions possibles depuis les états qui n'ont pas encore été marqués. On voit clairement que chaque transition amène dans un autre état qui n'est pas marqué. On a trouvé les classes d'équivalences :

$$1 \approx 6 \quad 2 \approx 5 \quad 3 \approx 4$$

Et l'automate minimal est :



2. Si on examine les transitions de l'automate, on voit qu'il n'y a pas d'états qui ne sont pas atteignables. Dès lors, on peut appliquer directement l'algorithme de minimisation :

1							
-	2						
-	-	3					
-	-	-	4				
-	-	-	-	5			
-	-	-	-	-	6		
-	-	-	-	-	-	7	
-	-	-	-	-	-	-	8

On marque toutes les couples d'états qui ont un état final et un autre non-final.

1							
-	2						
$\surd$	$\surd$	3					
$\surd$	$\surd$	-	4				
$\surd$	$\surd$	-	-	5			
$\surd$	$\surd$	-	-	-	6		
$\surd$	$\surd$	-	-	-	-	7	
$\surd$	$\surd$	-	-	-	-	-	8

On regarde les transitions possibles depuis les couples qui ne sont pas encore marqués. On met en évidence les états déjà marqués.

input b :	$\{6, 3\}$	$\rightarrow$	$\{3, 2\}$	$\surd$
input b :	$\{7, 3\}$	$\rightarrow$	$\{4, 2\}$	$\surd$
input b :	$\{4, 5\}$	$\rightarrow$	$\{2, 8\}$	$\surd$
input b :	$\{4, 6\}$	$\rightarrow$	$\{2, 3\}$	$\surd$
input b :	$\{4, 7\}$	$\rightarrow$	$\{2, 4\}$	$\surd$
input a :	$\{5, 8\}$	$\rightarrow$	$\{1, 5\}$	$\surd$
input a :	$\{5, 3\}$	$\rightarrow$	$\{1, 7\}$	$\surd$
input a :	$\{7, 8\}$	$\rightarrow$	$\{1, 5\}$	$\surd$
input a :	$\{6, 8\}$	$\rightarrow$	$\{2, 5\}$	$\surd$
input a :	$\{3, 4\}$	$\rightarrow$	$\{7, 6\}$	
input b :	$\{3, 4\}$	$\rightarrow$	$\{2, 2\}$	
input a :	$\{3, 8\}$	$\rightarrow$	$\{7, 5\}$	
input b :	$\{3, 8\}$	$\rightarrow$	$\{2, 1\}$	
input a :	$\{5, 7\}$	$\rightarrow$	$\{1, 1\}$	
input b :	$\{5, 7\}$	$\rightarrow$	$\{8, 4\}$	
input a :	$\{1, 2\}$	$\rightarrow$	$\{3, 8\}$	
input b :	$\{1, 2\}$	$\rightarrow$	$\{5, 7\}$	
input a :	$\{4, 8\}$	$\rightarrow$	$\{6, 5\}$	
input b :	$\{4, 8\}$	$\rightarrow$	$\{2, 1\}$	
input a :	$\{5, 6\}$	$\rightarrow$	$\{1, 2\}$	
input b :	$\{5, 6\}$	$\rightarrow$	$\{8, 3\}$	
input a :	$\{7, 6\}$	$\rightarrow$	$\{1, 2\}$	
input b :	$\{7, 6\}$	$\rightarrow$	$\{4, 3\}$	

Après la première étape on peut marquer neuf autres états :

1							
-	2						
$\surd$	$\surd$	3					
$\surd$	$\surd$	-	4				
$\surd$	$\surd$	$\surd$	$\surd$	5			
$\surd$	$\surd$	$\surd$	$\surd$	-	6		
$\surd$	$\surd$	$\surd$	$\surd$	-	-	7	
$\surd$	$\surd$	-	-	$\surd$	$\surd$	$\surd$	8

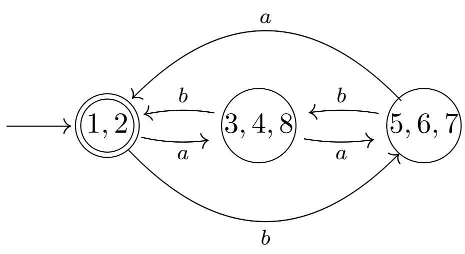
On examine les transitions possibles à partir des états qui ne sont pas encore marqués. On voit clairement que chaque transition mène dans un autre état qui n'est pas marqué. On a trouvé les classes d'équivalences :

$$3 \approx 4 \quad 3 \approx 8 \quad 5 \approx 7 \quad 1 \approx 2 \quad 4 \approx 8 \quad 5 \approx 6 \quad 7 \approx 6$$

Qui peuvent être réduites à :

$$1 \approx 2 \quad 3 \approx 4 \approx 8 \quad 5 \approx 6 \approx 7$$

Et l'automate minimal est :



2. Preuve de l'algorithme de minimisation

Démontrez que les relations $\approx$ et $\approx_m$ vue au cours coïncident, c'est à dire que $p \approx_m q \Leftrightarrow p \approx q$.

Aide : $p \approx_m q \Leftrightarrow p \approx q$ est équivalent à $\neg(p \approx_m q) \Leftrightarrow \neg(p \approx q)$.

Démontrez l'implication de gauche à droite ($\neg(p \approx_m q) \Rightarrow \neg(p \approx q)$) par induction sur les règles[1] définissant $\surd$.

Montrer l'implication de droite à gauche ($\neg(p \approx q) \Rightarrow \neg(p \approx_m q)$) revient en fait à démontrer la proposition suivante :

pour tout $n \in \mathbb{N}$: s'il existe x tel que $\mid x \mid = n$, $\hat{\delta}(p, x) \in F$ et $\hat{\delta}(q, x) \notin F$ alors $\{p, q\} \in \surd$.

Corrigé :

Les pas de l'algorithme peuvent être exprimés par les deux règles suivantes (cf. transparent 14, leçon 8) qui décrivent l'ensemble des pairs marquées ($\surd$). Bien entendu, l'application des règles continue seulement jusqu'à ce que il introduit des nouveaux pairs dans l'ensemble $\surd$.

$$(\textsc{Init}) \ \frac{p \in F \quad q \notin F}{\{p, q\} \in \surd}$$

$$(\textsc{Step}) \ \frac{a \in \Sigma \quad \delta(p, a) = p' \quad \delta(q, a) = q' \quad \{p', q'\} \in \surd}{\{p, q\} \in \surd}$$

Les pairs $\{p, q\}$ ne sont pas ordonnée, donc on a pas besoin de donner des règles symétriques.

On vous rappelle que :

$$\begin{array}{rll} P \Leftrightarrow Q & ssi & (P \Rightarrow Q) \wedge (Q \Rightarrow P) \\ & ssi & (\neg P \vee Q) \wedge (\neg Q \vee P) \end{array}$$

$$\begin{array}{rll} \neg\,(P \Leftrightarrow Q) & ssi & \neg\,((\neg P \vee Q) \wedge (\neg Q \vee P)) \\ & ssi & (\neg\,(\neg P \vee Q)) \vee (\neg\,(\neg Q \vee P)) \\ & ssi & (P \wedge \neg Q) \vee (Q \wedge \neg P) \end{array}$$

[1] Vue à la leçon 2, p. 19

On note que :

$$
\begin{array}{lll}
\neg\,(p \approx q) & ssi & \neg\left((\forall x \in \Sigma^*)\left(\hat{\delta}(p,x) \in F \Leftrightarrow \hat{\delta}(q,x) \in F\right)\right) \\
 & ssi & (\exists x \in \Sigma^*)\left(\neg\left(\hat{\delta}(p,x) \in F \Leftrightarrow \hat{\delta}(q,x) \in F\right)\right) \\
 & ssi & (\exists x \in \Sigma^*)\left(\left(\hat{\delta}(p,x) \in F \wedge \hat{\delta}(q,x) \notin F\right) \vee \left(\hat{\delta}(p,x) \notin F \wedge \hat{\delta}(q,x) \in F\right)\right)
\end{array}
$$

$\Rightarrow$
On démontre la proposition suivante par induction sur les règles (cf. transparent 19, leçon 2).
Pour tous $\{p,q\}$, si $\{p,q\} \in \surd$ alors il existe $x \in \Sigma^*$ tel que $\hat{\delta}(p,x) \in F$ et $\hat{\delta}(q,x) \notin F$ (ou vice-versa).
Pour tous $\{p,q\}$, $\{p,q\} \in \surd$ signifie que soit $p \in F \quad q \notin F$ (cas de base, application de la règle (INIT)) ou il existe un $a \in \Sigma$ tel que $\delta(p,a) = p' \quad \delta(q,a) = q'$ et $\{p',q'\} \in \surd$ (hypothèse d'induction, application de la règle (STEP)).

Cas (INIT)
Dans ce cas, $\{p,q\} \in \surd$ signifie que $p \in F$ et $q \notin F$, (ou vice-versa) dès lors, nous avons :

$$p = \hat{\delta}(p,\epsilon) \in F$$
$$q = \hat{\delta}(q,\epsilon) \notin F$$

Il existe donc un $x \in \Sigma^*$ (dans ce cas ϵ) tel que $\hat{\delta}(p,x) \in F$ et $\hat{\delta}(q,x) \notin F$ et on peut conclure que $p \not\approx q$.

Cas (STEP)
Dans ce cas, $\{p,q\} \in \surd$ signifie qu'il y a $a \in \Sigma$ tel que $\delta(p,a) = p'$ et $\delta(q,a) = q'$ et $\{p',q'\} \in \surd$. Et, par l'hypothèse d'induction sur $\{p,q\}$, nous savons que :

il existe $x' \in \Sigma^*$ tel que $\hat{\delta}(p',x') \in F$ et $\hat{\delta}(q',x') \notin F$ ou vice-versa

Prenons $x \in \Sigma^*$ tel que $x = ax'$, nous avons :

$$
\begin{array}{lll}
\hat{\delta}(p,x) & = & \hat{\delta}(p,ax') \\
 & = & \hat{\delta}\left(\delta(p,a),x'\right) \\
 & = & \hat{\delta}\left(p',x'\right) \in F
\end{array}
$$

et

$$
\begin{array}{lll}
\hat{\delta}(q,x) & = & \hat{\delta}(q,ax') \\
 & = & \hat{\delta}\left(\delta(q,a),x'\right) \\
 & = & \hat{\delta}\left(q',x'\right) \notin F
\end{array}
$$

(Le cas du vice-versa est symétrique)
Il existe donc un $x \in \Sigma^*$ (dans ce cas $x = ax'$) tel que $\hat{\delta}(p,x) \in F$ et $\hat{\delta}(q,x) \notin F$ (ou vice-versa) et on peut conclure que $p \not\approx q$.

Vu que le cas de bas et le cas d'induction sont vérifié, on peut terminer en disant que l'implication $\Rightarrow$ est vraie.

$\Leftarrow$
On doit démontrer que,
pour tout $n \in \mathbb{N}$: s'il existe x tel que $\mid x \mid = n$, $\hat{\delta}(p, x) \in F$ et $\hat{\delta}(q, x) \notin F$ alors $\{p, q\} \in \surd$.
Nous démontrons ça par induction naturelle sur la longueur de le mot x

Cas de Base
La longueur de x est zéro, à savoir $x = \epsilon$. Alors :

$$\hat{\delta}(p, \epsilon) = p \in F$$

$$\hat{\delta}(q, \epsilon) = q \notin F$$

(ou vice-versa)

Donc, par application de la règle (INIT), le pair $\{p, q\}$ est marqué dans l'algorithme, à savoir $\{p, q\} \in \surd$.

Cas d'Induction
La longueur de x est $n + 1$. On a comme hypothèse d'induction que, pour tout pair $\{p', q'\}$, s'il existe un mot de longueur n $x' \in \Sigma^*$ tel que $\hat{\delta}(p', x') \in F$ et $\hat{\delta}(q', x') \notin F$ alors $\{p', q'\}$ est marqué dans l'algorithme, à savoir $\{p', q'\} \in \surd$.
Alors, on peut décomposer x comme la concaténation de $a \in \Sigma$ et $x' \in \Sigma^*$, de longueur respectivement 1 et n, à savoir $x = ax'$.
Pour tout pair $\{p, q\}$, s'il existe un mot de longueur $n+1$ $x \in \Sigma^*$ tel que $\hat{\delta}(p, x) \in F$ et $\hat{\delta}(q, x) \notin F$ (ou vice-versa) on peut toujours écrire x comme $x = ax'$ et décomposer $\hat{\delta}(p, x) \in F$ dans $\hat{\delta}\left(\delta(p, a), x'\right) \in F$ et $\hat{\delta}(q, x) \notin F$ dans $\hat{\delta}\left(\delta(q, a), x'\right) \notin F$ (ou vice-versa). Si on pose $\delta(p, a) = p'$ et $\delta(q, a) = q'$ on voit que $\hat{\delta}(p', x') \in F$ et $\hat{\delta}(q', x') \notin F$.
Alors, par l'hypothèse d'induction, $\{p', q'\} \in \surd$.
On a que il existe un $a \in \Sigma$ tel que $\delta(p, a) = p'$ et $\delta(q, a) = q'$ et $\{p', q'\} \in \surd$, donc, par application de la règle (STEP), le pair $\{p, q\}$ est marqué dans l'algorithme, à savoir $\{p, q\} \in \surd$.