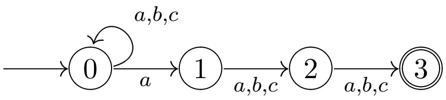
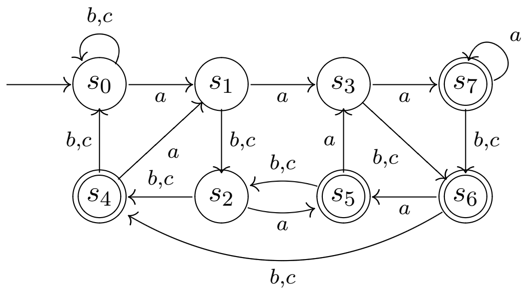
Automates & Calculabilité — Corrigé Série 4 — 10 nov 2003

1. Conversion d'un AFN en AFD

Convertir l'AFN (sur l'alphabet $\Sigma \stackrel{\text{def}}{=} \{a, b, c\}$) donné par le diagramme ci-dessous en AFD (en utilisant la construction vue au cours).



Solution :

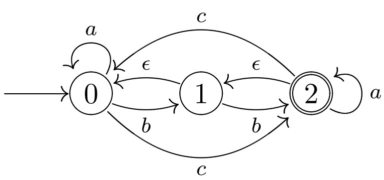


où

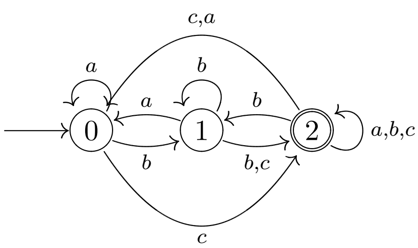
$$s_0 \stackrel{\text{def}}{=} \{0\} \quad s_1 \stackrel{\text{def}}{=} \{0,1\} \quad s_2 \stackrel{\text{def}}{=} \{0,2\} \quad s_3 \stackrel{\text{def}}{=} \{0,1,2\}$$
$$s_4 \stackrel{\text{def}}{=} \{0,3\} \quad s_5 \stackrel{\text{def}}{=} \{0,1,3\} \quad s_6 \stackrel{\text{def}}{=} \{0,2,3\} \quad s_7 \stackrel{\text{def}}{=} \{0,1,2,3\}$$

2. Élimination des ϵ-transitions

Transformer l'AFN$_\epsilon$ (sur l'alphabet $\Sigma \stackrel{\text{def}}{=} \{a, b, c\}$) donné par le diagramme ci-dessous en AFN ne contenant pas de ϵ-transitions (en utilisant l'ϵ-fermeture).



Solution : Nous avons que $C_\epsilon(0) = \{0\}$, $C_\epsilon(1) = \{0,1\}$ et $C_\epsilon(2) = \{0,1,2\}$. Les états accepteurs sont $\{\, q \in \{0,1,2\} \mid 2 \in C_\epsilon(q) \,\} = \{2\}$. L'automate résultant est



3. Explosion d'états.

Prouver que pour tout alphabet fini Σ avec $\|\Sigma\| > 1$ et tout $k > 1$ et $a \in \Sigma$, le langage $L_k \stackrel{\text{def}}{=} \{w\,a\,l_1\,l_2 \cdots l_{k-2} \mid w \in \Sigma^*, l_i \in \Sigma\}$ est tel que :

1. Il existe un AFN avec k états qui reconnaisse L_k.
2. Tout AFD reconnaissant L_k a au moins 2^{k-1} états.

Solution :

1. Nous définissons un AFN A_k, censé reconnaître L_k, comme

$$\begin{array}{rrcl} & A_k & \stackrel{\text{def}}{=} & (\{0, 1, \cdots, k-1\},\ \Sigma,\ \Delta_k,\ \{0\},\ \{k-1\}) \\ \text{où} & \Delta_k & \stackrel{\text{def}}{=} & \left\{ \begin{array}{rcll} (0, a) & \mapsto & \{0, 1\} & \\ (0, b) & \mapsto & \{0\} & \text{if } a \neq b \in \Sigma \\ (i, b) & \mapsto & \{i+1\} & \text{if } 0 < i < k-1 \text{ and } b \in \Sigma \\ (k-1, b) & \mapsto & \emptyset & \text{if } b \in \Sigma \end{array} \right. \end{array}$$

   – Pour prouver que A_k accepte L_k, nous prenons un mot quelconque $w\,a\,l_1\,l_2 \cdots l_{k-2} \in L_k$, et nous prouvons que $k-1 \in \widehat{\Delta}_k(\{0\}, wal_1l_2 \cdots l_k)$. Si c'est vrai A_k accepte L_k, comme $k-1$ est un état final.
   A forteriori, nous prouvons que $i+1 \in \widehat{\Delta}_k(\{0\}, wal_1l_2 \cdots l_i)$ pour tout $i \leq k-2$, par induction naturelle sur i.
   Pour $i = 0$ il faut prouver $1 \in \widehat{\Delta}_k(\{0\}, wa)$. Comme $\forall a \in \Sigma : 0 \in \Delta_k(0, a)$ nous avons $0 \in \widehat{\Delta}_k(\{0\}, w)$. Dès lors $1 \in \widehat{\Delta}_k(\{0\}, wa)$.
   Supposons que $i+1 \in \widehat{\Delta}_k(\{0\}, wal_1l_2 \cdots l_i)$ pour un $i < k-2$. Comme $\Delta_k(i+1, l_{i+1}) = \{i+2\}$ nous avons $i+2 \in \widehat{\Delta}_k(\{0\}, wal_1l_2 \cdots l_il_{i+1})$.
   Par induction, $i+1 \in \widehat{\Delta}_k(\{0\}, wal_1l_2 \cdots l_i)$ pour tout $i \leq k-2$. Alors $k-1 \in \widehat{\Delta}_k(\{0\}, wal_1l_2 \cdots l_k)$, et comme $k-1$ est un état final, A_k accepte L_k.

   – Pour prouver que A_k n'accepte que L_k, nous prenons un mot v accepté par A_k et prouvons que $v \in L_k$.
   Posons $v = v_1v_2 \cdots v_n$. Nous prouvons que pour tout $i \leq k-2$ nous avons $n \geq i+1$ et $k-i-1 \in \widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-i})$, par induction naturelle sur i.
   Posons $i = 0$. Comme A_k accepte v et $k-1$ est le seul état accepteur de A_k, nous avons $k-1 \in \widehat{\Delta}_k(\{0\}, v)$. Comme $k > 2$ alors $k-1$ n'est pas l'état initial 0, d'où $n \geq 1$.
   Supposons que $i < k-2$, $n \geq i+1$ et $k-i-1 \in \widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-i})$. Comme $k-i-1 \neq 0$, il n'est pas un état initial et alors $n \geq i+2$. En outre, $\Delta_k^{-1}(\{k-i-1\}) = \{(k-i-2,\ b) \mid b \in \Sigma\}$ et alors $k-i-1 \in \widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-i-1})$.
   Par induction, pour tout $i \leq k-2$ nous avons $n \geq i+1$ et $k-i-1 \in \widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-i})$.
   Comme un cas spécial, nous avons $n \geq k-1$ et $1 \in \widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-k+2})$.
   Par définition $\widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-k+2}) = \{\Delta_k(q, v_{n-k+2}) \mid q \in \widehat{\Delta}_k(\{0\}, v_1v_2 \cdots v_{n-k+1})\}$.
   Par la définition de Δ_k nous avons que $1 \in \Delta_k(q, v_{n-k+2})$ ssi $q = 0$ et $v_{n-k+2} = a$.
   En posant $w \stackrel{\text{def}}{=} v_1v_2 \cdots v_{n-k+1}$ et $l_i \stackrel{\text{def}}{=} v_{n-k+2+i}$ nous avons alors que $v = w\,a\,l_1\,l_2 \cdots l_{k-2} \in L_k$.

2. Étant donnés un automate déterministe $D_k \stackrel{\text{def}}{=} (Q_k, \Sigma, \delta_k, s, F)$ reconnaissant L_k, nous voulons montrer qu'il a au moins 2^{k-1} états. L'idée du preuve est de trouver un langage L'_k avec 2^{k-1} mots, tels que $\widehat{\delta}_k$ est injective sur ce langage. Dans ce cas, il existe un état dans Q_k pour chaque mot dans L'_k, d'où l'automate a au moins 2^{k-1} états.
   Pour la preuve, nous prenons $b \in \Sigma$ tel que $b \neq a$ et posons $L'_k \stackrel{\text{def}}{=} \{a, b\}^{k-1}$. Nous avons $\|L'_k\| = 2^{k-1}$. Posons $f : L'_k \to Q_k : v \mapsto \widehat{\delta}_k(0, v)$. Nous montrons que f est injective par contradiction.

Sinon, il existe $v, v' \in L'_k$ tels que $v \neq v'$ et $f(v) = f(v')$. Nous procédons à construire deux mots $v'\, b^{i-1}$ et $v\, b^{i-1}$ tels que l'un appartient à L_k et l'autre non. Notons que

$$\begin{aligned}
\widehat{\delta}_k(0, v'\, b^{i-1}) &= \widehat{\delta}_k(\widehat{\delta}_k(0, v'), b^{i-1}) \\
&= \widehat{\delta}_k(f(v'), b^{i-1}) \\
&= \widehat{\delta}_k(f(v), b^{i-1}) \\
&= \widehat{\delta}_k(\widehat{\delta}_k(0, v), b^{i-1}) \\
&= \widehat{\delta}_k(0, v\, b^{i-1})
\end{aligned}$$

Alors D_k accepte $v\, b^{i-1}$ ssi il accepte $v'\, b^{i-1}$.
Posons $v = v_1 v_2 \cdots v_{k-1}$ et $v' = v'_1 v'_2 \cdots v'_{k-1}$. Comme $v \neq v'$ il existe i tel que $0 < i < k$ et $v_i \neq v'_i$. Comme $v_i, v'_i \in \{a, b\}$, nous pouvons supposer que $v_i = a$ et $v'_i = b$ par symmetrie. Dès lors $v\, b^{i-1} = v_1 \cdots v_{i-1} a\, v_{i+1} \cdots v_{k-1} b \cdots b \in L_k$ mais $v'\, b^{i-1} = v'_1 \cdots v'_{i-1} b\, v'_{i+1} \cdots v'_{k-1} b \cdots b \notin L_k$, ce qui contredit que D_k reconnaît L_k.
Comme f est injective alors $\|Q_k\| \geq \|L'_k\| = 2^{k-1}$.