DM2 (éléments de réponses)

Préliminaires

Question 1. Supposons que $\mathcal{A}$ ne soit pas un automate standard. Cela signifie qu'une transition existe vers l'état initial q_0. Afin de construire un automate fini déterministe standard $\mathcal{A}'$ reconnaissant le même langage que $\mathcal{A}$, l'idée générale consiste à ajouter un état q_0', état initial de $\mathcal{A}'$, pour lequel n'existe aucune transition menant à q_0'. En outre, toutes les transitions issues de q_0 sont dupliquées depuis q_0'.
Plus formellement, soit q_0' un nouvel état n'appartenant pas à Q. On pose alors $Q' = Q \cup \{q_0'\}$, ensemble des états de $\mathcal{A}'$. Si $q_0 \in F$, on pose $F' = F \cup \{q_0'\}$; sinon, $F' = F$; cela permet de conserver le caractère acceptant de l'état initial. Enfin, la fonction de transition est modifiée en $\delta' = \delta \cup \{(q_0', x, q) \mid (q_0, x, q) \in \delta\}$.
L'automate $\mathcal{A}' = (\Sigma, Q', q_0', F', \delta')$ est standard et reconnaît le même langage de $\mathcal{A}$.

Langages locaux

Question 2. Soit L un langage local sur un alphabet Σ. Alors $P(L)$, $S(L)$ et $N(L)$ étant des langages finis, ce sont des langages réguliers. Par ailleurs, l'ensemble des langages réguliers est stable par union, produit et complémentation. Par conséquent, $P(L)\Sigma^*$, $\Sigma^* S(L)$, $\Sigma^* N(L) \Sigma^*$, puis $\hat{L} = (P(L)\Sigma^* \cap \Sigma^* S(L)) \smallsetminus (\Sigma^* N(L) \Sigma^*)$ sont aussi des langages réguliers. Enfin, puisque $L = \hat{L} \cup \{\varepsilon\}$ ou $L = \hat{L}$ selon que ε appartient ou non à L, L est un langage régulier.

Question 3. Notons $\Sigma = \{a, b, c\}$. On a $P = \{a\}$, $S = \{c\}$, $F = \{ab, bc, ca\}$, $N = \Sigma^2 \smallsetminus F$. Alors :

$$L_{ER}((abc)^*) \smallsetminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \smallsetminus (\Sigma^* N \Sigma^*)$$

Question 4. Notons $\Sigma = \{a, b\}$. On a $P = \{a, b\}$, $S = \{a\}$, $F = \{aa, ab, ba\}$, $N = \Sigma^2 \smallsetminus F$. Le mot $abab$ appartient au langage $(P\Sigma^* \cap \Sigma^* S) \smallsetminus (\Sigma^* N \Sigma^*)$ mais il n'appartient pas à $L_{ER}(a^* ba) \smallsetminus \{\varepsilon\}$. Le langage dénoté par $a^* ba$ n'est pas local.

Question 5. Sur l'alphabet $\Sigma = \{a, b\}$, considérons les langages $L_1 = \{ab\}$ et $L_2 = \{ba\}$. Avec $P_1 = \{a\}$, $S_1 = \{b\}$, $F_1 = \{ab\}$ et $N_1 = \Sigma^2 \smallsetminus F_1$, on a $L_1 = (P_1 \Sigma^* \cap \Sigma^* S_1) \smallsetminus (\Sigma^* N_1 \Sigma^*)$. Donc L_1 est local. Par raison de symétrie, il en est de même de L_2.
En revanche, $L_1 \cup L_2 = \{ab, ba\}$ n'est pas local. En effet, avec $P(L_1 \cup L_2) = \{a\}$, $S(L_1 \cup L_2) = \{b\}$, $F(L_1 \cup L_2) = \{ab, ba\}$ et $N = \Sigma^2 \smallsetminus F$, le mot aba appartient au langage $(P\Sigma^* \cap \Sigma^* S) \smallsetminus (\Sigma^* N \Sigma^*)$, ce qui établit que $L_1 \cup L_2$ n'est pas local.

Question 6. Sur l'alphabet $\Sigma = \{a, b\}$, reprenons le langage $L_1 = \{ab\}$ dont on a montré qu'il est local. Notons à présent $P_1 = P(L_1 \cdot L_1) = \{a\}$, $S_1 = S(L_1 \cdot L_1) = \{b\}$, $F_1 = F(L_1 \cdot L_1) = \{ab, ba\}$, $N_1 = \Sigma^2 \smallsetminus F_1$. Alors, le langage $(P_1 \Sigma^* \cap \Sigma^* S_1) \smallsetminus (\Sigma^* N_1 \Sigma^*)$ est infini ; c'est $L_{ER}((ab)^+)$. En particulier, il n'est pas égal à $L_1 \cdot L_1$. Par conséquent, $L_1 \cdot L_1$ n'est donc pas local.

Question 7. Soit L un langage régulier sur l'alphabet X et ϕ un morphisme de X^* vers Y^*. On définit par induction structurelle l'application $\hat{\phi}$ de l'ensemble des expressions régulières sur lui-même par :

- $\hat{\phi}(\varnothing) = \varnothing$
- $\hat{\phi}(\varepsilon) = \varepsilon$
- $\forall x \in X, \hat{\phi}(x) = \phi(x)$
- $\hat{\phi}(e + e') = \hat{\phi}(e) + \hat{\phi}(e')$
- $\hat{\phi}(e \cdot e') = \hat{\phi}(e) \cdot \hat{\phi}(e')$
- $\hat{\phi}(e^*) = \left(\hat{\phi}(e)\right)^*$

À un expression régulière e décrivant un langage L, $\hat{\phi}$ associe une expression régulière $\hat{\phi}(e)$ décrivant le langage $\phi(L)$. Avec le formalisme adopté ici, on a $\phi(L_{ER}(e)) = L_{ER}\left(\hat{\phi}(e)\right)$, soit $\phi \circ L_{ER} = L_{ER} \circ \hat{\phi}$.

Question 8. Soit $Y = Q \times \Sigma \times Q$ un alphabet (des transitions). Tout calcul sur $\mathcal{A}$ est un mot sur Y. Notons alors P l'ensemble des transitions qui partent de l'état initial, S l'ensemble des transitions qui mènent à un état final et F l'ensemble des couples $((q, x, q'), (q', x', q''))$ de transitions qui s'enchaînent. Enfin, notons $N = Y^2 \smallsetminus F$. Alors, l'ensemble des calculs réussis de $\mathcal{A}$ est exactement $(PY^* \cap Y^* S) \smallsetminus (Y^* N Y^*)$, ce qui montre que c'est un langage local.

Question 9. Soit ϕ le morphisme projectif qui, à la transition (q, x, q'), associe la lettre x qui étiquette cette transition. L'image par ϕ d'un calcul de $\mathcal{A}$ est l'étiquette de ce calcul. En particulier, l'image par ϕ de l'ensemble des calculs réussis de $\mathcal{A}$, qui est un langage local d'après la question précédente, est l'ensemble des étiquettes des calculs réussis de $\mathcal{A}$, autrement dit $L_{AF}(\mathcal{A})$.

Automates locaux

Question 10.
□ **10.1.** Soit L un langage local reconnu sur un alphabet Σ. Considérons l'automate $\mathcal{A}$ dont l'état initial est ε, dont les états acceptants sont les éléments de $S(L)$ plus éventuellement ε si ε appartient au langage et les transitions de deux types : (ε, a, a) avec $a \in P(L)$ et (x, y, y) avec $xy \in F(L)$. Alors, on vérifie que $L_{AF}(\mathcal{A}) \smallsetminus \{\varepsilon\} = (PX^* \cap X^*S) \smallsetminus X^*NX^*$.
□ **10.2.** Soit $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ un automate local. On ne restreint pas la généralité en supposant que chaque état autre de q_0 est une lettre de Σ qui étiquette les transitions menant à cet état. Notons $P = \{x \in \Sigma \,|\, (\varepsilon, x, x) \in \delta\}$, $S = F$, $F = \{xy \,|\, (x, y, y) \in \delta\}$ et $N = \Sigma^2 \smallsetminus F\}$. On a alors :

$$L_{AF}(\mathcal{A}) \smallsetminus \{\varepsilon\} = (P\Sigma^* \cap \Sigma^* S) \smallsetminus \Sigma^* N \Sigma^*$$

Question 11.
□ **11.1.** Soit $\mathcal{A}_1 = (X, Q_1, q_{01}, F_1, \delta_1)$ et $\mathcal{A}_2 = (Y, Q_2, q_{02}, F_2, \delta_2)$ deux automates locaux standards sur des alphabets disjoints qui reconnaissent L_1 et L_2. Sans perte de généralité, on peut supposer que Q_1 et Q_2 sont également disjoints. Soit alors q_0 un nouvel état. Posons alors :

$$\begin{aligned}
Q &= Q_1 \smallsetminus \{q_{01}\} \cup Q_2 \smallsetminus \{q_{02}\} \cup \{q_0\} \\
F &= \begin{cases} F_1 \cup F_2 & \text{si } q_{01} \notin F_1 \text{ et } q_{02} \notin F_2 \\ F_1 \smallsetminus \{q_{01}\} \cup F_2 \smallsetminus \{q_{02}\} & \text{sinon} \end{cases} \\
\delta &= \{(q, x, q') \in \delta_1 \,|\, q \neq q_{01}\} \cup \{(q, x, q') \in \delta_2 \,|\, q \neq q_{02}\} \\
&\quad \cup \{(q_0, x, q') \in \delta_1 \,|\, (q_{01}, x, q') \in \delta_1\} \cup \{(q_0, x, q') \in \delta_2 \,|\, (q_{02}, x, q') \in \delta_2\}
\end{aligned}$$

L'automate $\mathcal{A} = (X \cup Y, Q, q_0, F, \delta)$ est local et reconnaît $L_1 \cup L_2$.
□ **11.2.** Soit $\mathcal{A}_1 = (X, Q_1, q_{01}, F_1, \delta_1)$ et $\mathcal{A}_2 = (Y, Q_2, q_{02}, F_2, \delta_2)$ deux automates locaux standards sur des alphabets disjoints qui reconnaissent L_1 et L_2. Sans perte de généralité, on peut supposer que Q_1 et Q_2 sont également disjoints. Posons alors :

$$\begin{aligned}
Q &= Q_1 \cup Q_2 \smallsetminus \{q_{02}\} \\
F &= \begin{cases} F_2 & \text{si } q_{02} \notin F_2 \\ F_1 \cup F_2 \smallsetminus \{q_{02}\} & \text{sinon} \end{cases} \\
\delta &= \delta_1 \cup \{(q, x, q') \in \delta_2 \,|\, q \neq q_{02}\} \cup \{(q, x, q') \,|\, q \in F_1 \text{ et } (q_{02}, x, q') \in \delta_2\}
\end{aligned}$$

L'automate $\mathcal{A} = (X \cup Y, Q, q_{01}, F, \delta)$ est local et reconnaît $L_1 \cdot L_2$.
□ **11.3.** Soit $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ un automate local reconnaissant L. Notons $\delta' = \delta \cup \{(q, x, q') \,|\, q \in F \text{ et } (q_0, x, q') \in \delta\}$. L'automate $\mathcal{A} = (\Sigma, Q, q_0, F \cup \{q_0\}, \delta')$ reconnaît L^*.

Algorithme de McNaughton, Yamada et Glushkov

Question 12. On raisonne par induction structurelle.
Tout d'abord, $LER(\varnothing) = \varnothing$ est local ; il suffit de prendre $P = \varnothing$.
Ensuite, $LER(\varepsilon) = \{\varepsilon\}$ est local ; prendre encore $P = \varnothing$. Si $x \in \Sigma$, $L_{ER}(x) = \{x\}$. On prend $P = \{x\}$, $S = \{x\}$ et $N = X^2$.
Soient e et e' deux expressions régulières linéaires. Si $e + e'$ et $e \dots e'$ sont linéaires, alors $L_{ER}(e)$ et $L_{ER}(e')$ sont des langages locaux sur des alphabets distincts. Donc $L_{ER}(e + e') = L_{ER}(e) \cup L_{ER}(e')$ est local d'après le résultat de la question 12 et $L_{ER}(e \cdot e') = L_{ER}(e) \cdot L_{ER}(e')$ est local d'après le résultat de la question 13.
Enfin, $L_{ER}(e^*)$ est local d'après le résultat de la question 14.

Question 13. La réciproque est fausse. $L_{ER}(aa^*)$ est un langage local. On a $P = \{a\}$, $S = P$ et $F = \{aa\}$. Mais il ne peut pas être décrit par une expression régulière linéaire.

Question 14. On pose :

$$\lambda(\varnothing) = \varnothing \quad \lambda(\varepsilon) = \{\varepsilon\} \quad \forall x \in \Sigma, \lambda(x) = \varnothing \quad \lambda(e + e') = \lambda(e) \cup \lambda(e') \quad \lambda(e \cdot e') = \lambda(e) \cdot \lambda(e') \quad \lambda(e^*) = \{\varepsilon\}$$

Question 15. On pose :

$$\pi(\varnothing) = \varnothing \quad \pi(\varepsilon) = \varnothing \quad \forall x \in \Sigma, \pi(x) = \{x\} \quad \pi(e+e') = \pi(e) \cup \pi(e') \quad \pi(e \cdot e') = \pi(e) \cup (\lambda(e) \cdot \pi(e')) \quad \pi(e^*) = \pi(e)$$

Question 16. De la même façon, on pose :

$$\sigma(\varnothing) = \varnothing \quad \sigma(\varepsilon) = \varnothing \quad \forall x \in \Sigma, \sigma(x) = \{x\} \quad \sigma(e+e') = \sigma(e) \cup \sigma(e') \quad \sigma(e \cdot e') = \sigma(e) \cup (\sigma(e') \cup \lambda(e')) \quad \sigma(e^*) = \sigma(e)$$

Puis :

$$\phi(\varnothing) = \varnothing \quad \phi(\varepsilon) = \varnothing \quad \forall x \in \phi, \phi(x) = \{x\} \quad \phi(e+e') = \phi(e) \cup \phi(e')$$
$$\phi(e \cdot e') = \phi(e) \cup \phi(e') \cup (\sigma(e) \cdot \pi(e')) \quad \phi(e^*) = \phi(e) \cup (\sigma(e) \cdot \pi(e))$$

Question 17. Après linéarisation, l'expression régulière $((ab(ac)^* + ca)^*b)^*$ devient $e' = ((a_1b_1(a_2c_1)^* + c_2a_3)^*b_2)^*$. En appliquant l'algorithme de McNaughton-Yamada-Glushkov, on trouve :

$\pi(e') = \{b_2, c_2, a_1\}$
$\sigma(e') = \{b_2\}$
$\phi(e') = \{b_2b_2, b_2c_2, b_2a_1, a_3b_2, c_1b_2, b_1b_2, a_3c_2, c_1c_2, b_1c_2, a_3a_1, c_1a_1, b_1a_1, c_2a_3, b_1a_2, a_2c_1, c_1a_2, a_1b_1\}$

L'automate construit compte huit états (un pour chaque lettre de e' et l'état initial ε) et dix-sept transitions. (dessin à faire)

Question 18. L'automate fini $\mathcal{A}'$ construit à la troisième étape de l'algorithme de McNaughton-Yamada-Glushkov est local. $L_{ER}(e)$ est l'image de $L_{AF}(\mathcal{A}')$ par le morphisme $a_i \mapsto a$ de décodage.