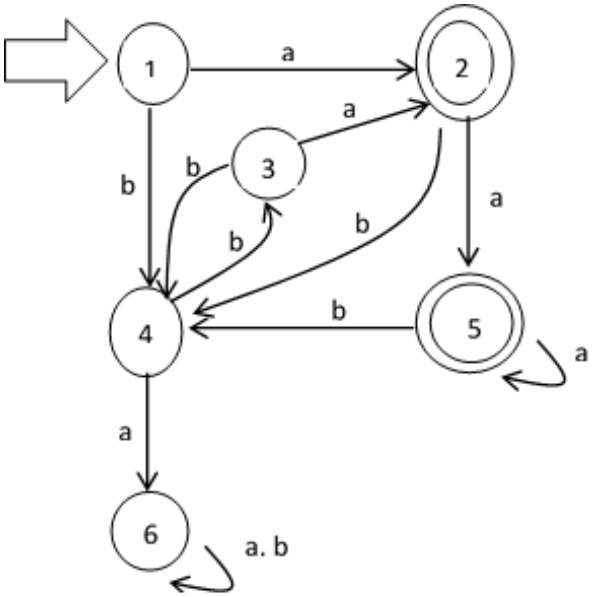
Institut des Sciences
Appliquées et Économiques
ISAE-Cnam Liban
Centre du Liban Associé au CNAM de Paris

Juillet-Durée:1h30
Partiel
2019-2020

Sujet coordonné par: J.Saab
Proposé pour les centres d'enseignement de:
Beyrouth-Baalbek-Tripoli-Bickfaya
Nahr Ibrahim
Langue de l'éxamen: Français

Partiel
Automates, Codes, Graphes- MVA004

1. (30points) L'alphabet étant $\Sigma = \{a, b\}$, on note L le langage reconnu par l'automate A défini par le diagrame



   (a) Donner la liste des mots de L de longueur inférieur ou égale à 4.
   (b) Écrire les équations du départ pour A.
   (c) Si on note les langages du départ D_1 à D_6, que vaut D_6. comparer D_1 et D_3, puis D_2 et D_5.
   (d) Résoudre le système précédent, et en déduire une expression régulière pour le langage L.
   (e) Donner le mot le plus court de chacun des langages D_1 à D_6. En déduire l'automate minimal B qui accepte le langage L.

---

2. (30points) Utiliser l'algorithme de Gluskov pour construire l'automate fini correspondant aux expressions régulières suivantes

   (a) $(a + b)^* c$
   (b) $a^*(\varepsilon + bb)a + \varepsilon$

---

3. (10points **bonus**) Quel est le langage généré par les grammaires suivantes $G = (T, N, S, R)$

   (a) $T = \{a, b\},\ N = \{S\},\ R = \{S \to aSbb|\varepsilon\}$
   (b) $T = \{a, b\},\ N = \{S, U\},\ R = \{S \to aUc;\ U \to aUb|\varepsilon\}$

---

4. (15 pts) Les affirmations suivantes sont-elles vraies? Justifier.

   (a) $baa \in a^*b^*a^*b^*$

   (b) $b^*a^* \cap a^*b^* = a^* \cup b^*$

   (c) $a^*b^* \cap c^*d^* = \phi$

   (d) $abcd \in (a(cd)^*b)^*$

   (e) $(u+v)^* = (u^* + v)^*$

   (f) $(u+v)^* = (u^*v^*)^*$

---

5. (25 pts) Soit $\Sigma = \{a, b\}$. Donner une expression rationnelle définissant le langage complémentaire du langage $L = b^*a^+b\Sigma^*$

Formulaire:

1 Système de départ:

   (a) A chaque état q_i on associe un langage de départ (inconnu) D_i

   (b) Si les symboles $\alpha_j,\ j = 1 \cdots k$, font passer q_i vers les états q_j,alors on établit l'équation

$$D_i = \begin{cases} \sum_{j=1}^{k} \alpha_j D_j & \text{si} \quad q_i \text{ n'est pas acceptant} \\ \sum_{j=1}^{k} \alpha_j D_j + \varepsilon & \text{si} \quad q_i \text{ est acceptant} \end{cases}$$

   (c) $L(A_d) = D_0$ le langage associé à l'état initial.

   (d) La résolution du système de départ se base sur la formule:

$$X = uX + v \text{ alors } X = \begin{cases} u^*v & \text{si} \quad \varepsilon \notin u \\ u^*(v+w) & \text{si} \quad \varepsilon \in u \end{cases} \text{ où } w \text{ est quelconque}$$

2. Algorithme de Gluskov pour la construction d'un AFD, A_d qui reconnait une expression régulière, sachant qu'une expression régulière est une chaîne finie de symboles de Σ dont certains ayant l'exposant $*$ ou $+$. (Par exemple aab^*ab^+)

   1-On linéarise l'expression régulière aab^*ab^+ sera $x_1x_2x_3^*x_4x_5^+$

   2-On déterminer les objets suivants:

   2-1- Premier: Ensemble des x_i pouvant commencer un mot (Premier $= \{x_1\}$)

   2-2- Dernier: Ensemble des x_i pouvant finir un mot (Dernier $= \{x_5\}$)

   2-3-Pour tout x_i on détermine Suivant (x_i) : Ensemble des x_j pouvant suivre x_i comme ds le tableau relatif à l'expression traitée

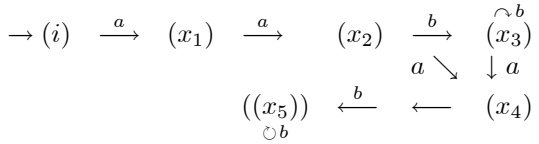
	Suivant
x_1	x_2
x_2	x_3, x_4
x_3	x_3, x_4
x_4	x_5
x_5	x_5

   3-On a la règle suivante: $\omega = \omega_1 \cdots \omega_n \in L(A)$ ssi $\omega_1 \in$ Premier, $\omega_n \in$ Dernier, $\omega_{i+1} \in$ Suivant (ω_i) $\forall i = 1...n-1$.

4-$A = (\Sigma, Q \cup \{i\}, \{i\}, F, \delta)$: On a ajouté un état initial $\{i\}$, $Q = \{x_i\}$, $F =$ Dernier et on ajoute à F l'etat $\{i\}$ lorsque $\varepsilon \in L$ et δ est donnée par

$$\begin{cases} \delta(i, x_k) & = & x_k & \text{si } x_k \in \text{Premier} \\ \delta(x_i, x_j) & = & x_j & \text{si } x_j \in \text{ Suivant}(x_i) \end{cases}$$

5- On remplace les etats x_i par des états q_i et les symboles de transitions x_i par les symboles qui leur sont équivalent dans l'expression régulière.



3. Construction d'un automate $\overline{A}$ reconnaissant $\overline{L} = \Sigma^* - L$ le complémentaire de $L = L(A)$

Il suffit de mettre les états acceptants dans le A.F.D. (sous la forme complète) A comme non acceptants dans $\overline{A}$ et les états non acceptants dans A comme des états acceptants dans $\overline{A}$

4 Minimisation ou réduction d'un automate A:

- On élimine les états qui ne reçoivent aucune transition
- On écrit E l'ensemble des états, comme l'union de deux sous ensembles, l'un contient les états acceptants et l'autre contient les non acceptants $E = E_1 \cup E_2$
- On cherche les états obtenus par action des symboles de Σ sur les états de E_1 puis de E_2 et on regroupe les états équivalents ensemble, sachant que q_1 est équivalent à q_2 si et seulement si $\forall \alpha_i \in \Sigma,\ \alpha_i(q_1)$ appartient au même sous ensemble E_1 ou E_2 que $\alpha_i(q_2)$.
- On reécrit comme union de sous ensembles E_1 et E_2 selon que les états de E_i sont ou ne sont pas stables par action des symboles de Σ

$$E_1 = E_{11} \cup E_{12} \cup \cdots ; \quad E_2 = E_{21} \cup E_{22} \cup \cdots$$

où les éléments de chaque sous ensemble sont équivalents

- On repète cette itération jusqu'à ne plus pouvoir ajouter de sous ensembles. Finalement $E = G_1 \cup G_2 \cup \cdots \cup G_k$
- L'automate minimal, ayant le même langage que A, a k états, $G_1, \cdots, G_k$, l'état initial étant celui qui contient l'état initial de A et tout état contenant un état acceptant de A est un état acceptant pour l'autonate minimal.

5. Minimisation ou réduction d'un automate A ($2^{\text{ième}}$ façon):

On peut réduire un automate en remarquant que deux états ayant le même langage dans le système de départ sont supposés être équivalents.