Corrigé CCP 2014 Partie II : Langages et automates

1. Montrons que $\hat{h}$ est un homomorphisme de langage. Pour cela on montre par récurrence sur n la propriété suivante:

   "Pour tous mots m_1, m_2 avec $len(m_1) = n$, $\hat{h}(m_1.m_2) = \hat{h}(m_1).\hat{h}(m_2)$."

   Initialisation : si $n = 0$, m_1 est le mot vide donc le résultat est immédiat.

   Hérédite : Supposons la propriété vraie au rang n. Soit m_1 un mot de taille $n+1$, que l'on décompose en $m_1 = x.\tilde{m}_1$ avec $\tilde{m}_1$ de taille n. Alors pour tout mot m_2, en utilisant l'hypothèse de récurrence et la définition de $\hat{h}$:

   $$\hat{h}(m_1.m_2) = \hat{h}(x.\tilde{m}_1.m_2) = h(x).\hat{h}(\tilde{m}_1.m_2) = h(x).\hat{h}(\tilde{m}_1).\hat{h}(m_2) = \hat{h}(m_1).\hat{h}(m_2).$$

   La propriété est donc vraie pour tout n, on a bien montré que $\hat{h}$ est un homomorphisme de langage. Montrons de plus qu'il est Λ-libre. Supposons que m est un mot non vide. Il peut alors se décomposer $m = x.\tilde{m}$. Alors $\hat{h}(m) = h(x).\hat{h}(\tilde{m})$. Comme h est à valeurs dans $Y^* \setminus \{\Lambda\}$, $h(x)$ est non vide donc $\hat{h}(m)$ aussi. Donc $\hat{h}$ est Λ-libre.

2. Montrons que $\widehat{h_{|X}} = h$. On procède de nouveau par récurrence sur $n = len(m)$.

   Initialisation : Si m est le mot vide, on a $\widehat{h_{|X}}(\Lambda) = h(\Lambda) = \Lambda$ car ce sont des homomorphismes.

   Hérédite : Supposons le résultat vrai au rang n. Soit m un mot de taille $n+1$ que l'on décompose en $m = x.\tilde{m}$ avec $\tilde{m}$ de taille n. Alors par définition de $\widehat{h_{|X}}$ et par hypothèse de récurrence:

   $$\widehat{h_{|X}}(m) = \widehat{h_{|X}}(x.\tilde{m}) = h_{|X}(x).\widehat{h_{|X}}(\tilde{m}) = h(x).h(\tilde{m}) = h(m).$$

3. On obtient $\tilde{h}(e) = 10.(010 + 10010)^*$.

4. On montre par induction structurelle sur l'expression régulière e que $\tilde{e}$ est une expression régulière.

   Cas de base:

   - si $e = \emptyset$ ou $e = \Lambda$, $\tilde{h}(e) = e$ est bien une expression régulière.

   - si $e = a \in X$, $\tilde{h}(a) = h(a) \in Y^*$ est bien une expression régulière (tout mot peut être identifié à une expression régulière par itération de la règle de concaténation).

   Supposons maintenant que le résultat est vrai pour e_1 et e_2 et montrons qu'il reste vrai pour $e_1 + e_2$, $e_1.e_2$ et e_1^*. En utilisant les définitions des expressions régulières et de $\tilde{h}$ on a:

   - $\tilde{h}(e_1 + e_2) = \tilde{h}(e_1) + \tilde{h}(e_2)$ est bien une expression régulière

   - $\tilde{h}(e_1.e_2) = \tilde{h}(e_1).\tilde{h}(e_2)$ est également une expression régulière

- $\tilde{h}(e_1^*) = \tilde{h}(e_1)^*$ est aussi une expression régulière.

Par induction structurelle, pour tout expression régulière e, $\tilde{h}(e)$ est une expression régulière.

5. On montre de nouveau le résultat par induction structurelle sur l'expression régulière e.

Cas de base:
- Si $e = \emptyset$, $L(\tilde{h}(e)) = \hat{h}(L(e)) = \{\}$.
- Si $e = \Lambda$, $L(\tilde{h}(e)) = \hat{h}(L(e)) = \{\Lambda\}$.
- Si $e = a \in X$, on commence par remarquer que $L(h(a)) = \{h(a)\}$ (attention $h(a)$ n'est pas forcément une lettre), on le montre par récurrence sur la longueur du mot et en utilisant la règle de concaténation. On a alors:

$$L(\tilde{h}(a)) = L(h(a)) = \{h(a)\} = \{\hat{h}(a)\} = \hat{h}(L(a)).$$

Supposons maintenant que le résultat est vrai pour e_1 et e_2 et montrons qu'il reste vrai pour $e_1 + e_2$, $e_1.e_2$ et e_1^*:

$$\begin{aligned}
L(\tilde{h}(e_1 + e_2)) &= L(\tilde{h}(e_1) + \tilde{h}(e_2)) \\
&= L(\tilde{h}(e_1)) \cup L(\tilde{h}(e_2)) \\
&= \hat{h}(L(e_1)) \cup \hat{h}(L(e_2)) \\
&= \hat{h}(L(e_1) \cup L(e_2)) \\
&= \hat{h}(L(e_1 + e_2))
\end{aligned}$$

$$\begin{aligned}
L(\tilde{h}(e_1.e_2)) &= L(\tilde{h}(e_1).\tilde{h}(e_2)) \\
&= \{m_1.m_2 | m_1 \in L(\tilde{h}(e_1)), m_2 \in L(\tilde{h}(e_2))\} \\
&= \{m_1.m_2 | m_1 \in \hat{h}(L(e_1)), m_2 \in \hat{h}(L(e_2))\} \\
&= \{\hat{h}(m_1).\hat{h}(m_2) | m_1 \in L(e_1), m_2 \in L(e_2)\} \\
&= \{\hat{h}(m_1.m_2) | m_1 \in L(e_1), m_2 \in L(e_2)\} \\
&= \hat{h}(\{m_1.m_2 | m_1 \in L(e_1), m_2 \in L(e_2)\}) \\
&= \hat{h}(L(e_1.e_2)).
\end{aligned}$$

$$
\begin{aligned}
L(\tilde{h}(e_1^*)) &= L(\tilde{h}(e_1)^*) \\
&= L(\tilde{h}(e_1))^* \\
&= (\hat{h}(L(e_1)))^* \\
&= \bigcup_{n\in\mathbb{N}} (\hat{h}(L(e_1)))^n \\
&= \bigcup_{n\in\mathbb{N}} \{\hat{h}(x_1)\dots\hat{h}(x_n) | x_1\dots x_n \in L(e_1)\} \\
&= \bigcup_{n\in\mathbb{N}} \{\hat{h}(x_1\dots x_n) | x_1\dots x_n \in L(e_1)\} \\
&= \hat{h}(\bigcup_{n\in\mathbb{N}} \{x_1\dots x_n | x_1\dots x_n \in L(e_1)\}) \\
&= \hat{h}(\bigcup_{n\in\mathbb{N}} L(e_1)^n) \\
&= \hat{h}(L(e_1)^*)
\end{aligned}
$$

Par induction structurelle, le résultat est donc vrai pour toute expression régulière e.

6. Soit L_X un langage régulier sur X. Il existe une expression régulière e telle que $L_X = L(e)$. Soit h un homomorphisme de langage. Alors par la question 2, $h = \widehat{h_{|X}}$ et par la question 5 :

   $$h(L_X) = \widehat{h_{|X}}(L(e)) = L(\widetilde{h_{|X}}(e)).$$

   Par la question 4, $\widetilde{h_{|X}}(e)$ est une expression régulière donc $h(L_X)$ est bien un langage régulier.

7. Le langage de l'automate est décrit par l'expression régulière $a.(aa^*ba + bb)^*.b$.

8. On montre le résultat par récurrence sur $n = len(m_1)$.

   Initialisation : si $m_1 = \Lambda$, $\delta^*(o,\Lambda) = \{o\}$ donc pour tous états o, d et tout mot m_2:

   $$\exists q \in Q, (q \in \delta^*(o,\Lambda) \wedge d \in \delta^*(q,m_2)) \iff d \in \delta^*(o,m_2)$$

   Hérédité : Supposons le résultat vrai au rang n. Soit m_1 un mot de taille $n+1$ que l'on décompose $m_1 = x.\tilde{m_1}$ avec $\tilde{m_1}$ de taille n. Alors en utilisant les définitions et l'hypothèse de récurrence, pour tous états o et d et tout mot m_2:

   $$
   \begin{aligned}
   d \in \delta^*(o, m_1.m_2) &\iff d \in \delta^*(o, x.\tilde{m_1}.m_2) \\
   &\iff \exists q_1 \in Q, (q_1 \in \delta(o,x) \wedge d \in \delta^*(q_1, \tilde{m_1}.m_2)) \\
   &\iff \exists q_1 \in Q, (q_1 \in \delta(o,x) \wedge \exists q_2 \in Q, (q_2 \in \delta^*(q_1,\tilde{m_1}) \wedge d \in \delta^*(q_2,m_2))) \\
   &\iff \exists q_2 \in Q, (\exists q_1 \in Q, (q_1 \in \delta(o,x) \wedge q_2 \in \delta^*(q_1,\tilde{m_1})) \wedge d \in \delta^*(q_2,m_2)) \\
   &\iff \exists q_2 \in Q, (q_2 \in \delta^*(o, x.\tilde{m_1}) \wedge d \in \delta^*(q_2,m_2)) \\
   &\iff \exists q_2 \in Q, (q_2 \in \delta^*(o, m_1) \wedge d \in \delta^*(q_2,m_2))
   \end{aligned}
   $$

9. On obtient l'automate $\hat{h}^{-1}(\mathcal{E})$ avec les états $\{A,B,C,D\}$, A comme état initial, D comme état final, et les transitions: $\delta(A,0) = \{D\}$, $\delta(B,0) = \{A\}$, $\delta(B,1) = \{D\}$, $\delta(C,0) = \delta(C,1) = \{A\}$, $\delta(D,1) = \{B\}$.

10. Le langage de l'automate $\hat{h}^{-1}(\mathcal{E})$ peut être décrit par l'expression régulière $0.(100 + 11)^*$. Alors le langage $\hat{h}(L(\hat{h}^{-1}(\mathcal{E})))$ est décrit par l'expression régulière $ab.(babab + bb)^*$ et on a l'inclusion stricte $\hat{h}(L(\hat{h}^{-1}(\mathcal{E}))) \subsetneq L(\mathcal{E})$. (par exemple le mot $aabab$ est dans $L(\mathcal{E})$ mais pas dans $\hat{h}(L(\hat{h}^{-1}(\mathcal{E}))))$.

11. On montre de nouveau le résultat sur $n = len(m)$.

    Initialisation : pour $n = 0$, $m = \Lambda$ et:

$$d \in \delta^*_{\hat{h}^{-1}}(o, \Lambda) \Leftrightarrow d = o \Leftrightarrow d \in \delta^*(o, \Lambda) = \delta^*(o, \hat{h}(\Lambda)$$

    Hérédité: supposons le résultat vrai au rang n. Soit m de taille $n+1$ que l'on décompose $m = x.\tilde{m}$ avec $\tilde{m}$ de taille n. Alors:

$$\begin{aligned}
d \in \delta^*(o, \hat{h}(m)) &\iff d \in \delta^*(o, \hat{h}(x.\tilde{m})) \\
&\iff d \in \delta^*(o, h(x).\hat{h}(\tilde{m})) \\
&\iff \exists q \in Q, (q \in \delta^*(o, h(x)) \wedge d \in \delta^*(q, \hat{h}(\tilde{m}))) \\
&\iff \exists q \in Q, (q \in \delta^*(o, h(x)) \wedge d \in \delta^*_{\hat{h}^*_{-1}}(q, \tilde{m})) \\
&\iff \exists q \in Q, (q \in \delta^*_{\hat{h}^{-1}}(o, x) \wedge d \in \delta^*_{\hat{h}^*_{-1}}(q, \tilde{m})) \\
&\iff d \in \delta^*_{\hat{h}^{-1}}(o, x.\tilde{m}) \\
&\iff d \in \delta^*_{\hat{h}^{-1}}(o, m).
\end{aligned}$$

12. Montrons que $L(\hat{h}^{-1}(\mathcal{A})) = \hat{h}^{-1}(L(\mathcal{A}))$. Pour tout mot m:

$$\begin{aligned}
m \in L(\hat{h}^{-1}(\mathcal{A})) &\iff \exists i \in I, \exists t \in T, t \in \delta^*_{\hat{h}^{-1}}(i, m) \\
&\iff \exists i \in I, \exists t \in T, t \in \delta^*(i, \hat{h}(m)) \\
&\iff \hat{h}(m) \in L(\mathcal{A}) \\
&\iff m \in \hat{h}^{-1}(L(\mathcal{A})).
\end{aligned}$$

13. Soit L_Y un langage régulier sur Y. Soit $\mathcal{A}$ un automate le reconnaissant. Soit h un homomorphisme de langage. Par la question 2, $h = \widehat{h_{|X}}$. On considère l'automate $\mathcal{E} = \widehat{h_{|X}}^{-1}(\mathcal{A})$. Par la question 12:

$$L(\mathcal{E}) = \widehat{h_{|X}}^{-1}(L(\mathcal{A})) = h^{-1}(L_Y),$$

    donc par le théorème de Kleene, comme $h^{-1}(L_Y)$ est reconnu par un automate, c'est un langage régulier.