semaine du 8 février 2020

université PARIS-SACLAY

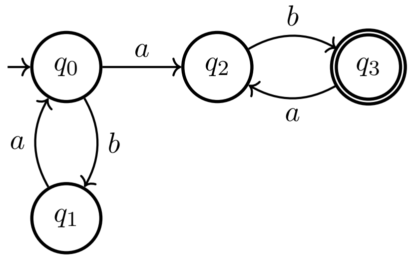
Automates et Applications
M1–MPRI Orsay

TD n° 4

1 Automates de Büchi et langages ω-réguliers

1. Donner un automate de Büchi acceptant les mots de la forme $(ba)^*(ab)^\omega$ sur l'alphabet $\{a, b\}$

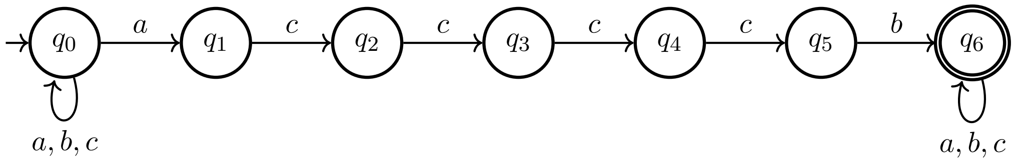
Réponse: Une astuce est de donner l'automate de mots reconnaissant : $(ba)^*(ab)^*$ et de s'assurer qu'une infinité de boucle dans les états acceptants reconnaissent bien le bon suffixe infini.



Un run infini d'un mot de $(ba)^*(ab)^\omega$ passera infiniment souvent par q_3.

2. Donner un automate de Büchi non-déterministe sur $\Sigma = \{a, b, c\}$ tel que pour tout mot de son langage, il existe une paire de lettres a, b séparées l'une de l'autre par 4 c.

Réponse: Il est agréable ici d'utiliser un automate non déterministe. On pourra donc l'écrire en trois partie :
— Une boucle qui consomme arbitrairement des a, b, c sur q_0 ainsi qu'un choix non déterministe pour sortir de $_0$ en lisant le bon a
— Une partie centrale lisant exactement le motif recherché
— Une boucle « finale » qui accepte n'importe quel suffixe, c'est à dire $(a|b|c)^\omega$.



3. Rappeler comment sont construits les langages ω-réguliers. Donner une condition nécessaire et suffisante sur un langage L non vide et ne contenant pas ϵ pour que L^ω soit fini. En déduire une condition nécessaire et suffisante pour qu'un langage ω-régulier soit fini.

Réponse: Les langages réguliers sont construits comme :
— La puissance ω d'un langage régulier de mots finis. Dans ce cas une condition suffisante pour que le langage ω-régulier soit fini est que le langage de mot soit un singleton. En effet, supposons que $A = \{a, b\}$. Alors A^ω contient :

$$\{a^n b^\omega \mid n \geq 0\}$$

qui est infini. Si $A = \{a\}$, alors le seul mot est a^ω On peut cependant relacher légèrement la restriction et demander que le langage soit de la forme A^* avec A un singleton. En effet, a, aa,

aaa ont tous la même expansion infinie. Pour que le langage ω-régulier soit infini, il faut donc qu'il contienne au moins deux mots u, v tels que $uv \neq vu$.

— La concaténation AB de deux langages est finie s'ils sont tous les deux finis, donc A doit être fini et non vide. Attention cependant, on a aussi un cas où A peut être infini et AB fini. En effet si $\forall u \in A, \exists w \in \Sigma^\omega$ tel que $uw \in B$, alors $|AB|$ est fini. En effet, Si on considère $A = a^*$ et $B = a^\omega$, alors $A = \{\epsilon, a, aa, aaa, aaaa, \ldots\}$ est bien infini, mais concaténer ces mots à a^ω ne crée pas de nouveaux mots.

— Pour l'union, la condition nécessaire et suffisante est que les deux langages soient finis.

4. Soit L un langage ω-régulier et A un langage régulier (de mots finis). On considère le langage $L' = \bigcup_{u \in A} \{w \mid uw \in L\}$. L' est-il ω-régulier ?

Réponse: Si on le formule différemment, L' est l'ensemble de tous les suffixes de mots dans L dont un préfixe est dans A. Prenons l'automate de $\mathcal{A}_A$ de A (il existe car A est régulier) et l'automate de Büchi $\mathcal{A}_L$ de L. Appliquons la construction du produit « simple » d'automates (sans tour). Distinguons l'ensemble des états $Q_{\mathsf{pref}} = \{(q, q') \mid q \in F_A\}$ c'est à dire l'ensemble des états contenant un état acceptant de $\mathcal{A}_A$. Considérons un mot $w = uv$ infini de L, dont un préfixe u est dans A. Considérons un run (infini) de ce mot : $(q_0, q'_0)(q_1, q'_1)\ldots$. Comme ce mot est reconnu par $\mathcal{A}_A$ le run atteint un état de Q_{pref}. Appelons (q_n, q'_n) cet état. On remarque que la suite du run $(q_n, q'_n)(q_{n+1}, q'_{n+1})\ldots$ reconnait v, qui est exactement un mot de L'. Donc un run acceptant (au sens du Büchi) qui commence en (q_n, q'_n) accepte un suffixe infini d'un mot de L dont le préfixe est dans A. Comme on peut construire un automate pour L', il est ω-régulier. L'automate final est $(Q, \Sigma, \delta, I, F)$ avec :

— $Q = Q_{\mathcal{A}_A} \times Q_{\mathcal{A}_L}$

— $I = \{(q, q') \mid q \in F_{\mathcal{A}_A}\}$

— $F = \{(q, q') \mid q' \in F_{\mathcal{A}_L}\}$

— $\delta((q, p), a) = \{(q', p') \mid q \in \delta_{\mathcal{A}_A}(q, a), p' \in \delta_{\mathcal{A}_L}(p, a)\}$

5. Montrer que le langage singleton contenant le mot infini $1\,01\,001\,0001\,00001\,\ldots\,0^n1\,\ldots$ n'est pas reconnaissable par un automate de Büchi.

Réponse: Considérons le mot $w = 1\,01\,001\,0001\,00001\,\ldots\,0^n1\,\ldots$. On constate que le $i^{\text{ème}}$ 1 est séparé du 1 suivant par i zéros.

Supposons que le langage est ω-régulier. Il existe alors un automate de Büchi le reconnaissant. On va raisonner par cas sur les portions de run d'automate comprise entre le premier passage et le second passage par un même état acceptant. Si on considère une tel portion de run :

$$\ldots q'_0, q'_1, \ldots, q'_k, q'_0 \ldots$$

avec q'_0 un état acceptant. Considérons la séquence des $(k+1)$ lettres lues lors de cette portion de run. On fait un raisonnement par cas sur le nombre de 1 parmi ces lettres :

— S'il n'y a aucun 1 (que des 0) cela signifie que l'on a :

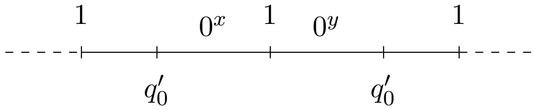
  — Un 1, que l'on suppose être le $i^{\text{ème}}$ du mot, suivi d'un certain nombre x de 0

  — puis une suite de 0 de taille $(k+1)$ (qui passe par notre portion de run)

  — puis une suite de 0 de taille y puis un 1 (le $(i+1)^{\text{ème}}$) du mot.

  On a aussi que $i = x + k + 1 + y$ (la distance entre les deux 1). Mais dans ce cas, l'automate reconnait aussi un mot passant deux fois par ce run. Dans un tel mot, le 1 en position i est séparé du suivant par : $x + 2 \times (k+1) + y \neq i$ zéros. Ce mot n'est pas dans le langage, contradiction.

— S'il y a un 1 (qu'on suppose le $i^{\text{ème}}$) dans le run cela signifie que l'on a :



  Si l'automate accepte un tel run, alors il accepte aussi un run passant 6 fois par q'_0 avant de poursuivre (3 tours de cette boucle). Il reconnait donc le mot $\ldots 10^x10^y\, 0^x10^y\, 0^x10^y \ldots$ Dans ce mots il y a deux paires de 1 séparés par le même nombre de zéros, ce qui n'est pas dans le langage, contradiction.

— S'il y a plusieurs 1 dans le run, c'est le même raisonnement que précédemment, si on fait un tour de plus, on a recréé une deuxième paire de 1 avec la même distance entre eux, ce qui est impossible.

On a supposé qu'il existait un automate qui reconnaissait *uniquement* le mot w, et on a à chaque fois vu que dans un tel automate, d'autre mots étaient forcément accepté. Il n'existe donc pas d'automate, le langage contenant uniquement ce mot n'est pas ω-régulier.

2 LTL

1. Pour chacune des questions ci-dessous, on donne un ensemble AP. Exprimer la formule demandée en LTL.
   (a) $AP = \{\mathsf{req}, \mathsf{ans}\}$ une requête (req) sera toujours satisfaite (aura une réponseans)
   (b) $AP = \{\mathsf{door}, \mathsf{code}\}$ À chaque fois que de l'on a fourni le (digi)code, la porte s'ouvre immédiatement après.
   (c) $AP = \{\mathsf{rouge}, \mathsf{orange}, \mathsf{vert}\}$ Lorsque le feu est vert, il passe au orange, puis au rouge (pas forcément immédiatement).

Réponse:

(a) Pour toutes les propriété de cette forme, il est utile de considérer une instance, puis ensuite d'utiliser $\mathcal{G}$ pour dire que cela est vrai à tout instant : $\mathcal{G}(\mathsf{req} \Rightarrow \mathcal{F}\mathsf{ans})$.

(b) Version plus contrainte de la précédente : $\mathcal{G}(\mathsf{code} \Rightarrow \mathcal{X}\mathsf{door})$

(c) Ici, utiliser $\mathcal{F}$ de façon indiscriminée pour dire que orange arrive après vert n'est pas judicieux, car alors on n'interdit pas au feu de passer au rouge entre les deux. On utilise donc $\mathcal{U}$:

$$\begin{aligned} &\mathcal{G}(\mathsf{vert}\,\mathcal{U}\,(\mathsf{orange}\,\mathcal{U}\,\mathsf{rouge})) \\ &\wedge \\ &\mathcal{G}((\mathsf{vert} \wedge \neg\mathsf{orange} \wedge \neg\mathsf{rouge}) \\ &\quad \vee(\neg\mathsf{vert} \wedge \mathsf{orange} \wedge \neg\mathsf{rouge}) \\ &\quad \vee(\neg\mathsf{vert} \wedge \neg\mathsf{orange} \wedge \mathsf{rouge})) \end{aligned}$$

La deuxième partie de la formule impose qu'on ne peut pas être dans deux couleurs en même temps.

2. **ATTENTION** dans cette section, la notation w^i ne signifie pas $wwwww\ldots$ répété i fois, mais le suffixe de w commençant à la position i. Cette surcharge de notation entre les deux théories (mots et langages de traces) est un peu malheureuse.

   Montrer les équivalences suivantes :

   — $\mathcal{F}\mathcal{F}\phi \equiv \mathcal{F}\phi$

Réponse: Rappelons la définition de $\mathcal{F}$: $\mathcal{F}\phi = \top\,\mathcal{U}\,\phi$. Supposons qu'un mot w satisfait $\top\,\mathcal{U}\,\phi$. C'est un mot tel que $\exists i,\ \text{tel que}\ w^i \vdash \phi$ et pour tout $k \leq i, \top$ (tautologie). Notons que si $w^i \vdash \phi$, alors $w^i \vdash \top\mathcal{U}\,\phi$, car il existe $i' = 0$ tel que $(w^i)^0 \vdash \phi$ (et il n'y pas de k plus petit que 0).
Dans l'autre sens, raisonnement symétrique. Si on supposes $\mathcal{F}\mathcal{F}\phi$ on peut trouver un suffixe w^i à partir du quel $\mathcal{F}\phi$ est vrai puis un préfixe $(w^i)^j$ à partir du quel ϕ est vrai. Comme il existe un préfixe $(w^i)^j = (w^{i+j})$ à partir duquel ϕ est vrai, $\mathcal{F}\phi$

— $\mathcal{F}\phi \equiv \phi \vee \mathcal{X}\mathcal{F}\phi$.

Réponse: Montrons que $\phi \lor \mathcal{XF}\phi \Rightarrow \mathcal{F}\phi$. Cela signifie plus précisement que pour tout w tel que $w \vdash \phi \lor \mathcal{XF}\phi$, alors il existe w' tel que $w' \vdash \mathcal{F}\phi$. De plus, $w' \vdash \mathcal{F}\phi$ s'il existe i tel que $w'^i \vdash \phi$. Par disjonction de cas sur $w' \vdash \phi \lor \mathcal{XF}\phi$

— Si $w \vdash \phi$, alors il existe $w' = w$ et $i = 0$ tel que $w'^0 \vdash \phi$, donc $w' \vdash \mathcal{F}\phi$.
— Si $w \vdash \mathcal{XF}\phi$ alors par définition de $\mathcal{X}$ $w^1 \vdash \mathcal{F}\phi$. On pose donc $w' = w$ et $i = 1$.

Montrons que $\phi \lor \mathcal{XF}\phi \Leftarrow \mathcal{F}\phi$. Cela Supposons qu'il existe w tel que $w \vdash \mathcal{F}\phi$. Montrons qu'il existe w' tel que $w' \vdash \phi \lor \mathcal{XF}\phi$. Comme on suppose $w \vdash \mathcal{F}\phi$, il existe i tel que $w^i \vdash \phi$. On peut encore raisonner par cas, cette fois sur i :

— si $i = 0$, alors $w^0 = w \vdash \phi$ donc $w \vdash \phi \lor \mathcal{XF}\phi$.
— si $i > 0$, alors on a $w^i \vdash \phi$. Comme $i > 0$, alors on a aussi que $w^1 \vdash \mathcal{F}\phi$. Donc par définition de $\mathcal{X}$, on a $w \vdash \mathcal{XF}\phi$.

— $\mathcal{X}(\phi\mathcal{U}\psi) \equiv (\mathcal{X}\phi)\mathcal{U}(\mathcal{X}\psi)$

Réponse: Montrons $\mathcal{X}(\phi\mathcal{U}\psi) \Rightarrow (\mathcal{X}\phi)\mathcal{U}(\mathcal{X}\psi)$ C'est à dire que pour tout $w \vdash \mathcal{X}(\phi\mathcal{U}\psi)$, il existe $w' \vdash (\mathcal{X}\phi)\mathcal{U}(\mathcal{X}\psi)$. Par définition de $\mathcal{X}$, on a $w^1 \vdash \phi\mathcal{U}\psi$, donc $\exists i$ tel que $(w^1)^i \vdash \psi$ et pour tout $k \leq i$, $(w^1)^k \vdash \phi$. En réarangent les exposants, on a $w^{i+1} \vdash \psi$ et pour tout $k \leq i$, $w^{k+1} \vdash \phi$. Donc, $w^i \vdash \mathcal{X}\psi$ et pour tout $k \leq i$, $w^k \vdash \mathcal{X}\phi$ et ainsi $(\mathcal{X}\phi)\mathcal{U}(\mathcal{X}\psi)$.

Montrons $\mathcal{X}(\phi\mathcal{U}\psi) \Leftarrow (\mathcal{X}\phi)\mathcal{U}(\mathcal{X}\psi)$. C'est à dire que pour tout $w \vdash (\mathcal{X}\phi)\mathcal{U}(\mathcal{X}\psi)$, il existe i tel que $w^i \vdash (\mathcal{X}\psi)$ et pour tout $k \leq i$, $w^k \vdash (\mathcal{X}\phi)$. Par définition de $\mathcal{X}$: $w^{i+1} \vdash \psi$ et pour tout $k \leq i$, $w^{k+1} \vdash \phi$. En réarrangeant les indices : $(w^1)^i \vdash \psi$ et pour tout $k \leq i$, $(w^1)^k \vdash \phi$. Ainsi, $(w^1) \vdash \phi\mathcal{U}\psi$. Et donc $w \vdash \mathcal{X}(\phi\mathcal{U}\psi)$.