Examen “Introduction à la vérification”

Master 1 Informatique, 2017–2018

24 avril 2018

Rédigez les 2 parties sur des copies séparées

Partie 1

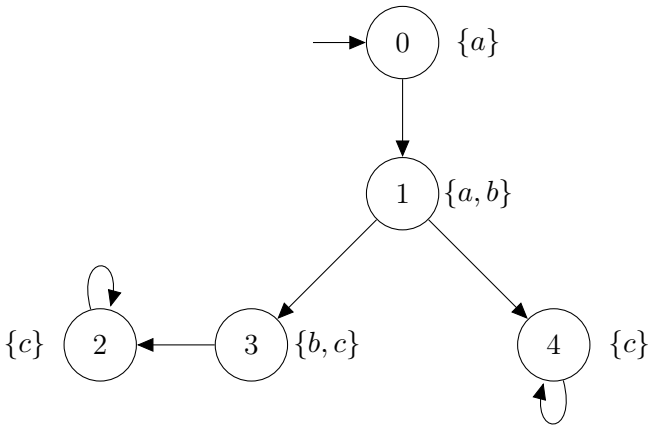
Exercice 1 Déterminez pour le système de transitions $\mathcal{S}$ et chacune des formules CTL ϕ_i qui suivent, si $\mathcal{S} \models \phi_i$.

- $\phi_1 = \mathbf{EF}\,\mathbf{AG}\,c$

  Pour ϕ_1 vous allez déterminer directement les états qui satisfont $\mathbf{AG}\,c$, puis ϕ_1, en justifiant votre réponse.

- $\phi_2 = \mathbf{A}(c\,\mathbf{U}\,((\mathbf{AF}\,c) \wedge \neg b))$

  Pour ϕ_2 vous allez appliquer l'algorithme de model-checking pour CTL vu en cours, en mettant d'abord la formule en forme normale existentielle, puis en calculant pour chaque sous-formule l'ensemble des états la satisfaisant. Vous allez noter la sous-formule $((\mathbf{AF}\,c) \wedge \neg b)$ par ϕ_3.



Exercice 2 Déterminez si les identités suivantes sont vraies, en justifiant votre réponse (a est une proposition atomique) :

- $\mathbf{EF}\,a = \mathbf{EF}\,\mathbf{EF}\,a$
- $\mathbf{AF}\,a = \mathbf{AF}\,\mathbf{AF}\,a$
- $\mathbf{EG}\,a = \mathbf{EG}\,\mathbf{EG}\,a$
- $\mathbf{EG}\,\mathbf{EX}\,a = \mathbf{EX}\,\mathbf{EG}\,a$
- $\mathbf{AG}\,\mathbf{AX}\,a = \mathbf{AX}\,\mathbf{AG}\,a$

Exercice 3 Dans cet exercice on cherche un algorithme pour calculer directement $\mathsf{Sat}(\mathbf{A}\,\phi)$, étant donné $U = \mathsf{Sat}(\phi)$ pour un système de transitions avec ensemble d'états S.

1. Pour $s \in S$ et $X, Y \subseteq S$ tels que $X \cap Y = \emptyset$, on définit $d(s, X, Y)$ comme la longueur *maximale* d'un chemin $s = s_1 \to s_2 \to \cdots \to s_n$, où $s_i \in X$ pour $i = 1, 2, \ldots, n-1$ et $s_n \in Y$. Le cas $n = 1$ signifie que $s \in Y$.

   Dans quel cas $d(s, X, Y) = \infty$?
2. Justifiez : si $s \models \mathbf{A}\,\phi$, alors $d(s, S \setminus U, U) \neq \infty$.
3. Soit $T \subseteq S$ un ensemble d'états qui satisfait les deux propriétés suivantes :

   (P1) $U \subseteq T$, et

   (P2) Si $s \in S$ est tel que $\mathrm{Post}(s) \subseteq T$, alors $s \in T$.

   Montrez que $\mathsf{Sat}(\mathbf{A}\,\phi) \subseteq T$.

   Indication : vous pouvez procéder par récurrence selon $d(s, S \setminus U, U)$.
4. Déduisez de la question précedente que $\mathsf{Sat}(\mathbf{A}\,\phi)$ est le plus petit sous-ensemble $T \subseteq S$ qui satisfait les deux propriétés (P1) et (P2).
5. Proposez un algorithme direct qui calcule $\mathsf{Sat}(\mathbf{A}\,\phi)$ à partir de $U = \mathsf{Sat}(\phi)$.

Exercice 4

1. Soient $\mathcal{S}_1, \mathcal{S}_2$ deux ST, $\mathcal{S}_i = (S_i, \to_i, I_i, \mathrm{AP}, L_i)$. Une simulation de $\mathcal{S}_1$ par $\mathcal{S}_2$ est une relation $R \subseteq S_1 \times S_2$ satisfaisant :
   - Pour tout $s_i \in I_1$ il existe $s_2 \in I_2$ t.q. $(s_1, s_2) \in R$.
   - Pour tous $(s_1, s_2) \in R$, $L(s_1) = L(s_2)$ et

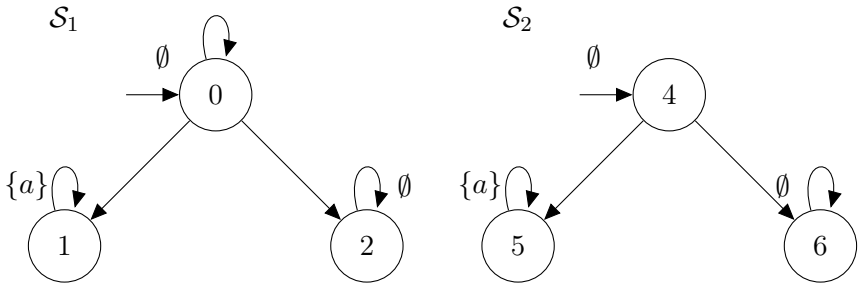
     $$\text{si } s_1 \to_1 s_1' \text{ alors il existe } s_2 \to_2 s_2' \text{ t.q. } (s_1', s_2') \in R$$

   $\mathcal{S}_1$ est simulé par $\mathcal{S}_2$ (noté par $\mathcal{S}_1 \preceq \mathcal{S}_2$) s'il existe une simulation de $\mathcal{S}_1$ par $\mathcal{S}_2$.
2. Le fragment universel de CTL, noté $\forall$CTL, est la restriction de CTL aux formules où les négations sont appliquées qu'aux propositions atomiques, et les opérateurs permis sont $\mathbf{AX}$, $\mathbf{A}\,\mathbf{U}$ et $\mathbf{A}\,\mathbf{R}$:

   $$\begin{aligned} \Phi &:= \mathrm{tt} \mid \mathrm{ff} \mid a \mid \neg a \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathbf{A}\,\phi \\ \phi &:= \mathbf{X}\,\Phi \mid \Phi_1\,\mathbf{U}\,\Phi_2 \mid \Phi_1\,\mathbf{R}\,\Phi_2 \end{aligned}$$

   L'opérateur $\mathbf{R}$ (release) est défini par $\Phi\,\mathbf{R}\,\Psi = \mathbf{G}\,\Psi \vee (\Psi\,\mathbf{U}(\Psi \wedge \Phi))$.

1. Justifiez pour les $\mathcal{S}_1, \mathcal{S}_2$ suivants : $\mathcal{S}_1 \not\preceq \mathcal{S}_2$.
2. Proposez une formule $\forall$CTL Φ telle que $\mathcal{S}_2 \models \Phi$ et $\mathcal{S}_1 \not\models \Phi$. Justifiez votre réponse.



Partie 2

On note $\mathrm{AP} = \{p, q, r, \ldots\}$ un ensemble de propositions atomiques et $\Sigma = 2^{\mathrm{AP}}$. Si α est une formule propositionnelle sur AP, on note $\Sigma_\alpha = \{a \in \Sigma \mid a \models \alpha\}$. Par exemple, une lettre appartient à $\Sigma_{p \wedge \neg q}$ si elle contient p mais pas q. Si $\mathrm{AP} = \{p, q, r\}$, on a ainsi $\Sigma_{p \wedge \neg q} = \{\{p\}, \{p, r\}\}$.

Exercice 5 Construire des automates de Büchi correspondant aux formules LTL suivantes. Vous pouvez utiliser des automates de Büchi généralisés (possédant plusieurs conditions d'acceptation). Il n'est pas demandé de suivre l'algorithme du cours. Justifier la correction des constructions proposées. Vous pouvez utiliser des sous-alphabets comme étiquettes pour représenter plusieurs transitions.

1. $\mathbf{G}(p \Rightarrow \mathbf{X}\,\mathbf{F}\,q)$.
2. $\mathbf{G}(p \Rightarrow (p\,\mathbf{U}\,q))$.
3. $\mathbf{G}(p \Rightarrow (p\,\mathbf{U}\,q)) \wedge \mathbf{G}(q \Rightarrow (q\,\mathbf{U}\,p))$.

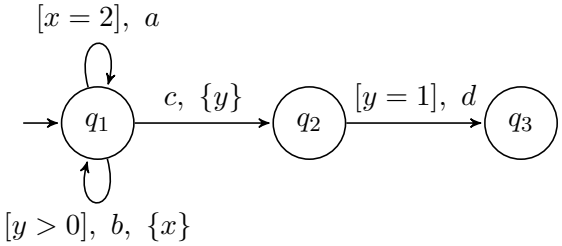
Exercice 6

1. Écrire des formules LTL décrivant les langages suivants :
   a. $\Sigma_p^n \Sigma_{\neg p}^\omega$, pour $n \geqslant 0$ fixé.
   b. $\Sigma_p^* \Sigma_{\neg p}^\omega$.
   c. $(\Sigma_p \Sigma_{\neg p})^\omega$.
2. (a) Soit u_n le mot infini suivant :
   $$u_n = \{p\}^{n+2} \varnothing^\omega.$$
   Montrer que pour toute formule LTL φ ayant au plus n occurrences de l'opérateur $\mathbf{X}$, on a :
   $$u_n, 0 \models \varphi \iff u_n, 1 \models \varphi.$$
   Vous pouvez raisonner par récurrence sur n, en distinguant plusieurs cas selon l'opérateur à la racine de l'arbre syntaxique de la formule φ (négation, disjonction, next ou until).

   (b) En utilisant la question précédente, prouver qu'il n'y a aucune formule LTL décrivant le langage $(\Sigma_p^2)^* \Sigma_{\neg p}^\omega$.
3. (a) On définit maintenant le mot v_n suivant :
   $$v_n = \{p\}^{n+2} (\{p\}\varnothing)^\omega.$$
   Montrer que pour toute formule LTL φ ayant au plus n occurrences de l'opérateur $\mathbf{X}$, on a :
   $$v_n, 0 \models \varphi \iff v_n, 1 \models \varphi.$$

   (b) En utilisant la question précédente, déterminer si $(\Sigma_p \Sigma)^\omega$ peut être exprimé en LTL. Justifier votre réponse.

Exercice 7 On considère l'automate temporisé suivant :



Les gardes sont indiquées entre crochets [ ], et les resets entre accollades {}.

1. La configuration $(q_3,\ x = 1,\ y = 2)$ est-elle accessible ? Si oui, donner un mot temporisé parvenant à cette configuration, et sinon, justifier. Même question pour la configuration $(q_3,\ x = 3,\ y = 2)$.
2. Calculer les régions de cet automate.
3. Calculer l'automate des régions et vérifier les réponses données à la question 1.

Exercice 8 Dans cet exercice, x, y représentent des horloges d'un automate temporisé. On appelle *garde diagonale* une expression de la forme, $x - y \bowtie c$, où

- x et y sont des horloges,
- $\bowtie$ est une relation de comparaison parmi $<, \leqslant, =, >\geqslant$,
- $c \in \mathbb{Z}$ est un entier.

Remarquez que les gardes diagonales ne sont pas, de base, autorisées dans la syntaxe des automates temporisés vue en cours. On veut cependant montrer qu'on n'augmente pas l'expressivité des automates temporisés si on se permet des gardes diagonales.

Autrement dit, étant donné un automate temporisé $\mathcal{A}$ ayant des gardes diagonales, on veut construire un automate temporisé $\mathcal{B}$ acceptant le même langage de mots temporisés, mais qui n'a pas de garde diagonale. L'idée est de supprimer les gardes diagonales *une par une* dans $\mathcal{A}$. On va donc fixer dans $\mathcal{A}$ une unique garde diagonale, que l'on suppose de la forme.

$$x - y \leqslant c.$$

Pour construire un automate $\mathcal{A}'$ équivalent à $\mathcal{A}$ mais qui n'a plus cette garde diagonale, on fait deux copies de $\mathcal{A}$: en notant Q l'ensemble d'états de $\mathcal{A}$ et Q' celui de $\mathcal{A}'$, on a $Q' = Q \times \{0, 1\}$. Un état (q, b) de Q' mémorise un état q de Q, et l'idée est que b vaudra 1 si $x - y \leqslant c$ et 0 si $x - y > c$.

On considère une transition $p \xrightarrow{g,\ a,\ R} q$ de $\mathcal{A}$, où g est la garde de la transition, la lettre a est le nom de l'action, et R désigne les resets (c'est-à-dire l'ensemble des variables remises à 0).

1. On suppose d'abord que la garde g ne contient pas la garde diagonale $x - y \leqslant c$. Décrire les transitions correspondantes de $\mathcal{A}'$, en distinguant 4 cas, suivant que $x \in R$ ou non, et suivant que $y \in R$ ou non.
2. Même question lorsque g contient la garde $x - y \leqslant c$ que l'on veut supprimer.
3. Comment adapter les questions précédentes lorsque $\bowtie$ est une autre relation que $\leqslant$?
4. Quelle est la taille de $\mathcal{A}'$ en fonction de celle de $\mathcal{A}$, dans le cas le pire ? En déduire la taille de $\mathcal{B}$, l'automate obtenu en ayant supprimé *toutes* les gardes diagonales, dans le cas le pire.