IGL502/752 – Techniques de vérification et de validation

Université de Sherbrooke

Examen final

Enseignant: Michael Blondin
Date: vendredi 17 décembre 2021
Durée: 3 heures

Directives:

- Vous devez répondre aux questions dans le **cahier de réponses**, et *non* sur ce questionnaire;
- **Une seule feuille** de notes au format 8½" × 11" est permise;
- **Aucun matériel additionnel** (notes de cours, fiches récapitulatives, etc.) n'est permis;
- **Aucun appareil électronique** (calculatrice, téléphone, montre intelligente, etc.) n'est permis;
- Vous devez donner **une seule réponse** par sous-question;
- L'examen comporte **6 questions** sur **3 pages** valant un total de **50 points**;
- La correction se base notamment sur la **clarté**, l'**exactitude** et la **concision** de vos réponses, ainsi que sur la **justification** pour les questions qui en requièrent une.

Question 1: logique temporelle linéaire (LTL)

Soient $AP \coloneqq \{p, q\}$ et les formules LTL suivantes sur AP, où « $\oplus$ » dénote l'opération « OU exclusif »:

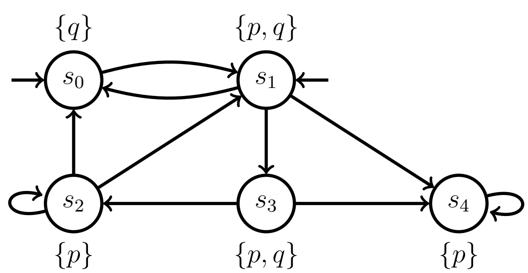
$$\varphi_1 \coloneqq \neg\mathsf{FG}(p \wedge q) \qquad \varphi_2 \coloneqq p \;\mathsf{U}\; (p \oplus q) \qquad \varphi_3 \coloneqq \mathsf{XX}q \vee \mathsf{G}(p \to \mathsf{F}\neg p)$$

(a) Pour chaque formule φ_i, donnez un mot σ_i qui la satisfait et qui ne satisfait pas les deux autres, c.-à-d. 6 pts

$$\begin{array}{lll} \sigma_1 \models \varphi_1 & \sigma_1 \not\models \varphi_2 & \sigma_1 \not\models \varphi_3, \\ \sigma_2 \not\models \varphi_1 & \sigma_2 \models \varphi_2 & \sigma_2 \not\models \varphi_3, \\ \sigma_3 \not\models \varphi_1 & \sigma_3 \not\models \varphi_2 & \sigma_3 \models \varphi_3. \end{array}$$

(b) Donnez un mot σ qui satisfait à la fois φ_1, φ_2 *et* φ_3, c.-à-d. tel que $\sigma \models \varphi_1$, $\sigma \models \varphi_2$ et $\sigma \models \varphi_3$. 2 pts

(c) Pour chaque formule φ_i, dites si la structure de Kripke $\mathcal{T}$ ci-dessous satisfait φ_i. Justifiez. 3 pts



Question 2: langages ω-réguliers

(a) Donnez une expression ω-régulière ***ou*** un automate de Büchi pour ce langage: 3 pts

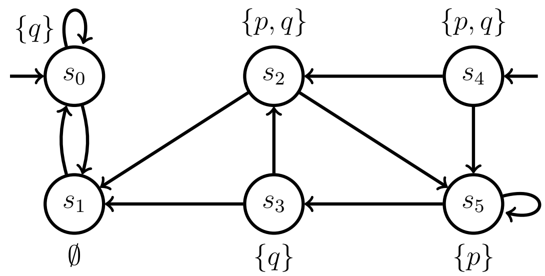
$$L \coloneqq \{\sigma \in \{a,b,c\}^\omega : \overbrace{[\forall i \in \mathbb{N}\ \exists j \geq i : \sigma(j) = a]}^{\sigma \text{ contient une infinité de } a} \wedge \underbrace{[\forall i, k \in \mathbb{N} : (\sigma(i) = a \wedge i < k \wedge \sigma(k) = a) \to |\{j \in [i..k] : \sigma(j) = b\}| \bmod 3 = 0]}_{\text{le nombre de } b \text{ entre chaque paire de } a \text{ est un multiple de } 3}\}.$$

(b) Donnez un automate de Büchi $\mathcal{B}$ tel que $\mathcal{L}(\mathcal{B}) = [\![\mathsf{X}(p \;\mathsf{U}\; (\mathsf{G}q)) \wedge \mathsf{GF}p]\!]$ sur alphabet $\Sigma \coloneqq \{\emptyset, \{p\}, \{q\}, \{p,q\}\}$. 3 pts

Question 3: logique temporelle arborescente (CTL) et vérification symbolique

Rappel: l'abbréviation « BDD » réfère à « diagramme de décision binaire (ordonné et réduit) ».

Supposons que chaque état de la structure de Kripke $\mathcal{T}$ ci-dessous soit codé par la représentation binaire de son indice: $s_0 = 000$, $s_1 = 001$, $s_2 = 010$, $s_3 = 011$, $s_4 = 100$ et $s_5 = 101$ (les autres chaînes sont invalides).



(a) Pour chaque formule Φ ci-dessous, donnez l'ensemble $[\![\Phi]\!]$ des états de $\mathcal{T}$ qui satisfont Φ, et dites si $\mathcal{T} \models \Phi$. 6 pts

(i) $\exists \mathsf{X} \forall \mathsf{G} \neg p$ (ii) $\forall (p \mathbin{\mathsf{U}} q)$ (iii) $\forall \mathsf{G} \exists \mathsf{F}(p \vee q)$

(b) Donnez un BDD qui représente les états initiaux I, l'ensemble $[\![p]\!]$ et l'ensemble $[\![q]\!]$. Indiquez clairement quels sommets du BDD correspondent à ces trois ensembles. 4 pts

Remarque: il n'est pas obligatoire d'appliquer un algorithme.

(c) Expliquez comment vérifier algorithmiquement si $\mathcal{T} \models p \wedge q$ à partir du BDD construit en (b). 2 pts

Question 4: systèmes à pile

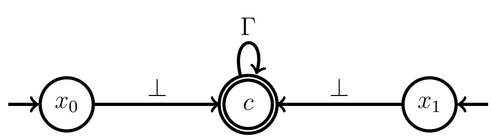
Considérons ce programme constitué de deux fonctions et d'une variable booléenne globale x:

```
     bool x ∈ {faux, vrai}

     foo(bool y):                   bar(bool z):
f0:    tant que x ∧ y:          b0:   x = x ⊕ z
         bar(¬y)

f1:    assert(¬x)
```

Remarque: l'absence d'étiquette dans le corps de la boucle « tant que » est volontaire; supposez que le corps est exécuté au même moment qu'une évaluation à « vrai » de la condition en f_0.

(a) Modélisez le programme avec un système à pile $\mathcal{P}$. 3 pts

(b) Construisez partiellement un $\mathcal{P}$-automate $\mathcal{B}$ qui accepte $\mathrm{Pre}^*(\mathit{Conf}(\mathcal{A}))$, où $\mathcal{A}$ est ce $\mathcal{P}$-automate: 3 pts



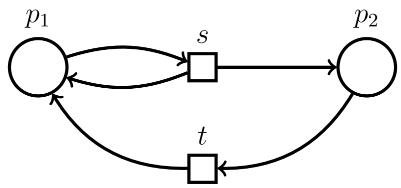
Plus précisément, ajoutez *au moins trois nouvelles transitions* à $\mathcal{A}$ en exécutant l'algorithme de saturation vu en classe. Au moins deux de ces transitions doivent être obtenues *sans* utiliser une transition de $\mathcal{P}$ étiquetée par une règle de la forme « *lettre* $\rightarrow \varepsilon$ ».

Rappel: Γ *est l'alphabet de* $\mathcal{P}$.

(c) Supposons que l'on ait complété $\mathcal{B}$ en (b). Si $(f_0, y_i) \in \mathit{Conf}(\mathcal{B})$, que peut-on conclure sur l'assertion? 1 pt

Question 5: réseaux de Petri

Soit le réseau de Petri $\mathcal{N} = (P, T, F)$ suivant:



(a) Dessinez un graphe de couverture qui débute en $\boldsymbol{m} \coloneqq (0, 1)$. Décrivez l'ensemble des marquages couvrables à partir de $\boldsymbol{m}$. 2,5 pts

(b) Exécutez l'algorithme arrière afin de déterminer l'ensemble des marquages qui peuvent couvrir $\boldsymbol{m}' \coloneqq (0, 1)$, c'est-à-dire $\uparrow\text{Pre}^*(\uparrow\boldsymbol{m}')$. 3 pts

(c) Dites lesquels de ces marquages peuvent couvrir $\boldsymbol{m}' = (0, 1)$. Justifiez brièvement. 1,5 pts

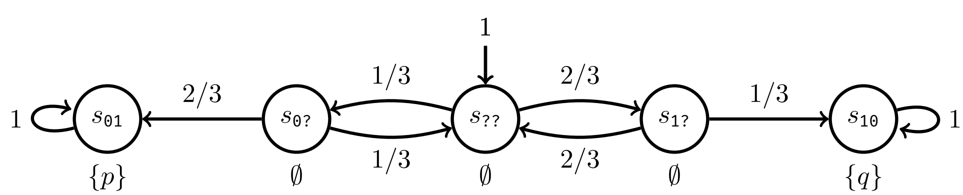
$$\boldsymbol{m}_0 \coloneqq (0, 1), \qquad \boldsymbol{m}_1 \coloneqq (3, 0), \qquad \boldsymbol{m}_2 \coloneqq (0, 0).$$

Question 6: chaînes de Markov

Considérons une pièce de monnaie biaisée qui retourne pile avec probabilité $1/3$ et face avec probabilité $2/3$. Cet algorithme cherche à simuler une pièce non biaisée à l'aide d'une pièce biaisée:

répéter
  choisir un bit x à pile ou face avec la pièce biaisée
  choisir un bit y à pile ou face avec la pièce biaisée
jusqu'à $x \neq y$
si $x = 0$ **alors retourner** *pile*
sinon retourner *face*

L'algorithme peut être modélisé à l'aide de la chaîne de Markov $\mathcal{M}$ ci-dessous, où p et q correspondent respectivement à « pile » et « face ». On aimerait donc que $\mathcal{M}$ satisfasse la propriété PCTL $\varphi \coloneqq \mathcal{P}_{=1/2}(\mathsf{F}\, p) \land \mathcal{P}_{=1/2}(\mathsf{F}\, q)$.



(a) La chaîne de Markov $\mathcal{M}$ satisfait φ. Expliquez pourquoi. 4 pts

Remarque: avec cette information, les calculs devraient être plutôt simples.

(b) Quelle est la valeur de $\mathbb{P}(s_{??} \models \mathsf{G}(\neg p \land \neg q))$? Justifiez. 1,5 pts

(c) Est-ce que $\mathcal{M}$ satisfait $\mathcal{P}_{\geq 1/3}(\mathsf{X}\,\mathcal{P}_{>0}(\mathsf{X}\, p))$? Justifiez. 1,5 pts