TD de Model Checking

24 octobre 2008

Modélisation des systèmes réactifs

Exercice 1 (Exemple de l'ascenceur.)**.** *Le système de contrôle d'un ascenceur (pour 3 étages) est défini par :*

- *le contrôleur garde en mémoire l'étage courant et l'étage cible.*
- *en mode actif, quand l'étage cible est atteint, les portes s'ouvrent et le contrôleur passe en mode attente.*
- *en mode actif, quand l'étage cible est plus élevé que l'étage courant, le contrôleur fait s'élever l'ascenceur.*
- *en mode actif, quand l'étage cible est moins élevé que l'étage courant, le contrôleur fait descendre l'ascenceur.*
- *en mode attente, il se peut que quelqu'un entre dans l'ascenceur et choisisse un nouvel étage cible. L'ascenceur ferme alors les portes et redevient actif.*
- *initialement, l'ascenceur est à l'étage 0 et en mode attente.*

Questions : 1. Proposez une machine à états modélisant le contrôle de l'ascenceur (définition formelle et dessin). 2. Définissez et dessinez le système de transitions correspondant (en vous limitant aux configurations accessibles depuis l'état initial). 3. Est-ce que les portes peuvent s'ouvrir quand l'ascenceur est actif ?

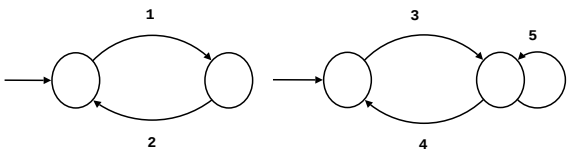
Exercice 2. *Soit un système de transitions $S = \langle Q, T, \rightarrow \rangle$ et $q_0 \in Q$ une configuration initiale. Montrez que :*

1. *l'ensemble d'accessibilité $post^*(q_0)$ est le plus petit invariant de S contenant q_0 ;*
2. *il existe $k \in \mathbb{N}$ tel que $post^*(q_0) = \bigcup_0^k post^i(q_0)$.*

Exercice 3 (Co-accessibilité)**.** *Nous avons vu comment vérifier l'invariance et l'accessibilité à partir du calcul des états accessibles ("calcul en avant"). On peut aussi vérifier ces propriétés en calculant "en arrière". On définit informellement $pre(q)$ comme les configurations à partir desquelles on peut atteindre q_0.*

- *définir formellement la relation pre (s'inspirer de post)*
- *On appelle ensemble de co-accessibilité de q l'ensemble $pre^*(q)$. Que représente-t-il intuitivement ?*
- *Donner un algorithme pour calculer $pre^*(q)$. Justifiez la terminaison.*
- *Comment vérifier l'invariance et l'accessibilité à partir de la co-accessibilité ?*
- *Montrer qu'il existe $k \in \mathbb{N}$ tel que $pre^*(q) = \bigcup_0^k pre^i(q)$.*

Exercice 4 (Concurrence)**.** *Soit les machines concurrentes suivantes.*



Quelles sont les transitions du système concurrent dans les cas suivants :

1. *sémantique synchrone ;*
2. *sémantique asynchrone (1) ;*
3. *sémantique asynchrone (2) ;*
4. *sémantique synchrone + synchronisation entre 1 et 3 ;*
5. *sémantique asynchrone (2) + synchronisation entre 1 et 3 ;*

Quel est le lien entre vecteur de synchronisations et rendez-vous ?

Exercice 5 (Sûreté)**.** *On se donne un système de transitions S dont certaines transitions sont distinguées et correspondent à des opérations d'aquisition de verrou (`lock`), de rendu de verrou (`unlock`), de lecture (`read`) et d'écriture (`write`). On se donne la propriété φ suivante :*

Si on ne regarde que les `lock` et `unlock` : `unlock` est toujours précédé directement de `lock` ET une suite arbitraire de `read`,`write` est toujours précédée directement d'un `lock` .

Questions :

1. *Est-ce que φ est une propriété de sûreté ? Sinon modifiez là en conséquence.*
2. *Écrivez un automate observeur pour vérifer φ (ou sa modification) et expliquez la nouvelle propriété à vérifier.*
3. *On considère maintenant la spécification :*

   Si on ne regarde que les `lock` et `unlock` : `unlock` est toujours précédé directement de `lock` `lock` est toujours suivi directement de `unlock`, ET une suite arbitraire de `read`,`write` est toujours précédée directement d'un `lock` et fermée directement par un `unlock`.

   Est-ce toujours une propriété de sûreté ?

Exercice 6 (**)**.** *Comment calculer $post_F^*$ et pre_F^* dans le cas d'un système de transitions avec hypothèses d'équité sur le modèle ?* (voir poly pour les définitions, la fin du cours peut aider).

Logiques temporelles

Exercice 7. *La vivacité est-elle de la sûreté ? Justifiez.*

Exercice 8. *Quelques petits exercices sur les connecteurs temporels :*
- $\mathbf{F}p$ *est-il vrai si p vrai tout de suite dans l'état courant ?*
- $\mathbf{G}p$ *est-il vrai si p faux dans l'état courant et vrai partout ailleurs ?*
- $p\mathbf{U}q$ *est-il vrai si p faux et q vrai dans l'état courant ?*
- $p\mathbf{U}q$ *est-il vrai si q est toujours faux, et p toujours vrai ?*

Exercice 9. *Dessinez des dépliages sur lesquels vous illustrerez les propriétés* **EX**, **AX**, **EU**, **AU**.

Exercice 10. *Exprimer les propriétés suivantes :*

1. *Tous les états satisfont p.*
2. *On peut atteindre p par un chemin où q est toujours vrai.*
3. *Quelquesoit l'état, on finit par revenir à l'état initial $init$.*
4. *Quelquesoit l'état, on peut revenir à l'état initial $init$.*
5. *Absence de deadlock (partiel).*

Exercice 11. *On va voir que certains connecteurs sont redondants.*
- *Exprimer $\mathbf{G}p$ avec les connecteurs $\neg, \mathbf{F}$ et p.*
- *Exprimer $\mathbf{F}p$ grâce au connecteur $\mathbf{U}$.*
- *Peut-on exprimer $\mathbf{X}$ en fonction des autres connecteurs ?*

– *Peut-on exprimer* $\mathbf{U}$ *en fonction des autres connecteurs ?*

Exercice 12 (Autres connecteurs.)**.** *On va définir quelques connecteurs additionels utiles.*

1. *Définir la relation* $\models$ *pour les connecteurs additionnels suivants :*

– $p\mathbf{W}q$ *(weak until) : signifie que p est vrai jusqu'à ce que q soit vrai, mais q n'est pas forcément vrai à un moment. Dans ce cas, p reste vrai tout le long du chemin.*
– $\mathbf{F}^{\infty}p$ *(infiniment souvent) : p est infiniment vrai au long de l'exécution.*
– $\mathbf{G}^{\infty}p$ *(presque toujours) : à partir d'un moment donné, p est toujours vrai.*
– $p\ \mathbf{U}_{\leq \mathbf{k}}q$ *(bounded until) : p vrai jusqu'à ce que q soit vrai, et q vrai dans au plus k observations.*
– $p\mathbf{R}q$ *(release) : q est vraie jusqu'à (et inclus) le premier état où p est vraie, sachant que p n'est pas forcément vraie un jour.*

2. *On va maintenant faire le lien entre ces connecteurs et les anciens.*

– *Exprimer* $\mathbf{F}^{\infty}$, $\mathbf{G}^{\infty}$, $\mathbf{W}$, $\mathbf{U}_{\leq \mathbf{k}}$ *par des connecteurs de basiques de LTL.*
– *Exprimer* $\mathbf{U}$ *dans LTL-*$\mathbf{U}$*+*$\mathbf{W}$.

Exercice 13. *Parmi les opérateurs suivants, lesquels correspondent plutôt à des propriétés de sûreté ?*
$\mathbf{X}, \mathbf{F}, \mathbf{G}, \mathbf{U}, \mathbf{W}, \mathbf{U}_{\leq \mathbf{k}}, \mathbf{F}^{\infty}, \mathbf{G}^{\infty}$.

Exercice 14. *Exprimer en langage naturel les propriétés suivantes.*

– $\mathbf{AG}(emission \rightarrow \mathbf{F}reception)$
– $\mathbf{AF}^{\infty}ok \rightarrow \mathbf{G}(emission \rightarrow \mathbf{F}reception)$

Exercice 15. *Exprimer toutes les propriétés de la section 3.1 en tenant compte des quantificateurs de chemin.*

Exercice 16 (CTL*)**.**

1. *Montrer que* $\vee$, $\neg$, $\mathbf{X}$, $\mathbf{U}$ *et* $\mathbf{E}$ *suffisent à exprimer les autres connecteurs.*
2. *Montrez que si on ajoute* $\mathbf{R}$*, on peut restreindre* $\neg$ *aux propositions atomiques.*

Exercice 17 (CTL)**.** *Montrer que p,* $\vee$, $\neg$, $\mathbf{EX}$, $\mathbf{EG}$ *et* $\mathbf{EU}$ *suffisent à exprimer les autres connecteurs. Montrer ensuite que p,* $\wedge$, $\neg$, $\mathbf{EX}$, $\mathbf{AU}$ *et* $\mathbf{EU}$ *suffisent aussi.*

Exercice 18. *Ce n'est pas parcequ'une formule n'est pas syntaxiquement dans CTL qu'il n'y a pas de formule CTL équivalente. Tranformez les formules suivantes en formules CTL :* $\mathbf{E}(p \wedge \mathbf{F}q)$, $\mathbf{AGF}p$.
Peut-on exprimer les notions suivantes en CTL : $\mathbf{AW}$, $\mathbf{EW}$, $\mathbf{AU}_{\leq \mathbf{k}}$, $\mathbf{EU}_{\leq \mathbf{k}}$ *?*
Peut-on exprimer les notions suivantes en CTL : $\mathbf{AG}^{\infty}$, $\mathbf{EG}^{\infty}$ *?*
Équité : peut-on exprimer $\mathbf{AF}^{\infty}\varphi$ *en CTL ? et* $\mathbf{EF}^{\infty}\varphi$ *?*

Exercice 19 (Comparaisons de LTL, CTL, CTL*)**.** *Quand une logique est incluse dans une autre, dîtes pourquoi. Quand deux logiques sont distinctes, trouvez une formule expressible dans l'une et pas dans l'autre.*

Exercice 20 (*)**.** *Montrez que* $\mathbf{U}$ *n'est pas associatif.*

Exercice 21 (Opérateurs du passé (**))**.** *On va définir des opérateurs du passé.*

1. *Modifier la définition de la logique et de la sémantique pour prendre en compte les opérateurs du passé* $\mathbf{X}^{-1}$,$\mathbf{F}^{-1}$, $\mathbf{G}^{-1}$ *et* $\mathbf{U}^{-1}$.
2. *Montrer que LTL+(opérateurs du passé) est équivalent à LTL.*
3. *Comparez la concision des deux logiques.*

Exercice 22 (Théorème de Kamp (*))**.** *Ce résultat est le sens (facile) du théorème de Kamp, qui établit que LTL a même pouvoir d'expression que la logique monadique du premier ordre à un successeur.*

On se donne un ensemble AP de prédicats atomiques $P : \mathbb{N} \rightarrow \mathcal{B}$ *et un ensemble Var de variables. On considère la logique monadique du premier ordre à un successeur, définie par :*

. $t := 0 | v \in Var | t + 1$

. *atome* : := $t \geq t | t = t | P(t), P \in AP$
. f : := $f \vee f | f \wedge f | \neg f | \exists v, f |$ *atome*

La logique est interprétée sur $\mathbb{N}$. *En considérant que* $P(t)$ *signifie que la propriété* P *est vrai au temps* t (= P *vrai à la* t*-ième étape du chemin), donner une traduction (récursive) des formules LTL* φ *en formules* $\widetilde{\varphi}$ *de logique monadique ayant même signification.*

Exercice 23 (Équivalence comportementale (*).)**.** *Soit deux structures de Kripke* $\mathcal{M}_1$ *et* $\mathcal{M}_2$ *et* φ *une propriété LTL. On suppose que* $\mathcal{M}_1 \models \varphi$. *Que peut-on dire si :*

1. $\mathcal{L}(\mathcal{M}_1) = \mathcal{L}(\mathcal{M}_2)$ *?*
2. $\mathcal{L}(\mathcal{M}_2) \subseteq \mathcal{L}(\mathcal{M}_1)$ *?*
3. $\mathcal{L}(\mathcal{M}_1) \subseteq \mathcal{L}(\mathcal{M}_2)$ *?*

On suppose maintenant que φ *est une propriété CTL. Reprendre la question 1.*

Exercice 24 (**)**.** *Montrer que le problème du model checking de LTL se ramène au problème de la validité de LTL. Plus précisément, on se donne une formule LTL* φ *et une structure de Kripke* $\mathcal{M} = \langle Q, \rightarrow, P, l, s_0 \rangle$. *On va construire une formule* φ'' *telle que* $\mathcal{M}, s_0 \models \varphi$ *ssi* φ'' *est valide. Pour cela on va construire une formule* φ' *telle que* $\sigma \models \varphi'$ *ssi* $\sigma \in \mathcal{L}(\mathcal{M})$. *On commence par rajouter une variable propositionnelle* p_{s_i} *pour chaque état* s_i *de* $\mathcal{M}$.

1. *Construisez les formules suivantes :* $PROP_{s_i}$ *qui indique les propriétés atomiques vérifiées par* s_i *et* $NEXT_{s_i}$ *qui mime la relation de transition de* $\mathcal{M}$.
2. *Servez-vous des résultats précédents pour construire* φ'.
3. *Concluez en construisant la formule* φ'' *cherchée à partir de* φ' *et* φ.

Exercice 25 (Bisimulation et simulation (*))**.** *à faire.*

Model checking

Exercice 26 (MC CTL)**.** *Modifier l'algorithme pour gérer tous les cas suivants :* $\neg p$, $\wedge$, $\mathbf{AX}$, $\mathbf{AG}$, $\mathbf{AF}$, $\mathbf{EF}$, $\mathbf{AR}$, $\mathbf{ER}$, $\mathbf{AW}$, $\mathbf{EW}$, $\mathbf{AU}_{\leq \mathbf{k}}$, $\mathbf{EU}_{\leq \mathbf{k}}$.

Exercice 27. *Écrire des automates de Büchi sur l'alphabet* $\{a, b\}$ *reconnaissant les langages suivants :* a^w, $b^* a (a, b)^w$.

Exercice 28. *Comment tester l'appartenance d'un mot au langage d'un automate de Büchi ? Comment tester le vide d'un automate de Büchi ?*

Exercice 29. *Transformez les propriétés de chemin suivantes en automates de Büchi sur alphabet* $\{p, q, \neg p, \neg q\}$ *:* $p, \neg p, \mathbf{X}p, \mathbf{F}p, \mathbf{G}p, p\mathbf{U}q, p\mathbf{W}q, \mathbf{F}^\infty p, \mathbf{G}^\infty p, p\mathbf{U}_{\leq \mathbf{3}} q$.

Exercice 30. *La complémentation des automates de Büchi est très coûteuse. Proposer une manière de s'en passer.*

Exercice 31 (Preuve de MC CTL (*))**.** *Faire les preuves de correction et de complexité de l'algorithme de model checking de CTL.*

Exercice 32 (Fair CTL (*))**.** *Terminer la preuve de correction. Notamment expliquez pourquoi on peut utiliser* FAIR-MARKING-EG *pour étiqueter* $\mathbf{EG}$*true* en sémantique fair *(point 2), et prouver les équivalences données pour passer de* $\models_F$ *à* $\models$. *Exprimer pour tous les connecteurs CTL la relation* $\mathcal{M}, s \models_F \varphi$ *en fonction de* $\models$ *et fair.*

Enfin essayer de réexprimer l'algorithme général plus simplement, en ramenant $\mathcal{M}, s \models_F \varphi$ *à* $\mathcal{M}', s \models \varphi$, *où* $\mathcal{M}'$ *est une autre structure de Kripke et* φ *est la même formule, sans fair.*

Exercice 33 (*). *Comment calculer l'union et l'intersection d'automates de Büchi ?*

Exercice 34 (Model checking de CTL* (*)). *Nous allons voir comment adapter les algorithmes vus en cours pour obtenir une procédure de model checking pour CTL*. Dans la suite, nous noterons $LTL_\forall$ la logique LTL standard. La notation vient de ce qu'une formule LTL de type $\mathbf{A}\varphi_p$ (φ_p sans quantificateur) est vraie sur une structure de Kripke $\mathcal{M}$ ssi tous les chemins de la structure satisfont la formule φ_p. Nous introduisons $LTL_\exists$ qui est une variante où $\mathbf{A}$ est remplacée par $\mathbf{E}$. Ainsi $\mathcal{M} \models_\exists \mathbf{E}\varphi_p$ ssi il existe un chemin de $\mathcal{M}$ satisfaisant la formule φ_p.*

1. *Montrer comment adapter l'algorithme de model checking de $LTL_\forall$ pour $LTL_\exists$.*
2. *Montrer comment modifier l'algorithme de model checking de $LTL_\forall$ pour marquer tous les états d'une structure de Kripke vérifiant une proriété de type $\mathbf{A}\varphi_p$. (actuellement, l'algorithme se contente de vérifier que l'état initial satisfait la propriété).* On évitera une solution du type : on relance l'algorithme pour chaque état.
3. *En déduire un algorithme de model checking pour CTL*, utilisant les procédures de marquages des états pour $LTL_\forall$ et $LTL_\exists$, et le principe de marquage récursif de CTL.*

Exercice 35 (Automates de Büchi et LTL (*)). *à faire*

Exercice 36 (Automates de Büchi généralisés (*)). *à faire*

Exercice 37 (Automates de Büchi déterministes (*)). *à faire*

Exercice 38 (CTL et point fixe (*)). *à faire*

Rappels de logique

Exercice 39 (Logique des propositions).

1. *Dites pour chaque interprétation $\mathcal{I}$ si elle est un modèle de la formule $A \vee (\neg B)$:*
   $\mathcal{I}_1 : (A, B) \longrightarrow (0, 0)$, $\mathcal{I}_2 : (A, B) \longrightarrow (1, 1)$, $\mathcal{I}_3 : (A, B) \longrightarrow (1, 0)$.
2. *Que dire des formules suivantes (satisfaisable, valide, contradictoire) : $A \wedge \neg A$, $A \vee \neg A$, $A \vee B$*
3. *Ajouter à la logique les connecteurs $\rightarrow$, $\leftrightarrow$ et `xor`.*
4. *Exprimer ces connecteurs en fonction des anciens.*
5. *Exprimer $\vee, \top, \bot$ en fonction de $\wedge, \neg$.*
6. *Montrer que tous les connecteurs peuvent s'obtenir à partir de $\neg p \wedge \neg q$.*

Exercice 40. *Quel lien y a-t-il entre satisfaisabilité de f et validité de $\neg f$?*

Exercice 41. *On définit la relation $\equiv$ sur les formules logiques par $\varphi_1 \equiv \varphi_2$ ssi φ_1 et φ_2 ont les mêmes modèles. Donnez une définition formelle de "ont les mêmes modèles". Quel est le lien entre $\varphi_1 \equiv \varphi_2$ et $\varphi_1 \leftrightarrow \varphi_2$?*

Exercice 42 (**). *Soit F un ensemble fini de formules de logique classique propositionnelle sur des propositions atomiques $p_1, \ldots, p_n$. À partir de quelle valeur de $|F|$ est-on sûr d'avoir au moins deux $\varphi_1, \varphi_2 \in F$ telles que $\varphi_1 \equiv \varphi_2$?*

Exercice 43 (Forme normale (*)). *Montrer que toute formule φ peut se mettre sous une forme $\bigvee_i \bigwedge_j \bar{p}_i$, où $\bar{p}_i$ vaut soit p_i soit $\neg p_i$.*

Exercice 44 (QBF (*)). *On appelle QBF (*Quantified boolean Formulas*) la logique des propositions à laquelle on ajoute les quantificateurs $\exists$ et $\forall$. On pourra ainsi écrire des formules comme : $\exists x, x \wedge y$.*

1. *Définissez $\models$ pour ces nouveaux opérateurs.*
2. *Montrer que toute formule φ de QBF peut se traduire en une formule $\widetilde{\varphi}$ de logique des propositions.*
3. *Quel lien y a-t-il entre $|\varphi|$ et $|\widetilde{\varphi}|$?*

Algorithmique

Exercice 45. *Prouvez la correction de l'agorithme de Kosaraju. Pour cela, vous procéderez en deux phases. D'abord (1) montrez que si x, y sont dans la même SCC alors ils sont dans le même arbre calculé par la DFS sur G_r. Puis (2) montrez le sens inverse.*