

TD Master Réseaux, SPEC, 2008-09

Partie vérification sur modèles

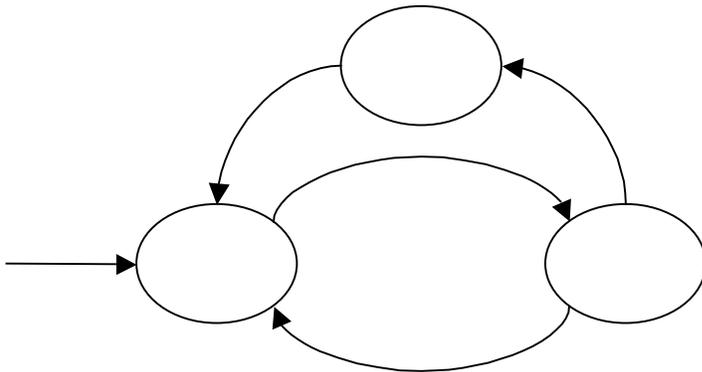
J. Leneutre

Exercice 1 : Vérification sur modèles de PLTL (examen 2008-09, semestre 1)

1. On considère la formule Φ de PLTL définie par : $\Phi \equiv (GF p) \Rightarrow (GF q)$

- a- Traduire en langage naturel la formule Φ .
- b- A quelle classe de propriété Φ appartient-elle ?
- c- Peut-on formuler une propriété équivalente à Φ dans CTL ? (justifier)
- d- Construire l'automate de Büchi reconnaissant la formule $\neg \Phi$.

2. On considérera dans la suite de cette exercice la structure de Kripke S définie sur l'ensemble de propositions $\{p, q\}$ par le schéma suivant :



- a- Construire le produit en S et l'automate de Büchi de la question 1.d.
- b- La formule Φ est-elle satisfaite par S ? (justifier en utilisant la question 2.a)

Exercice 2 : logique temporelle PLTL (examen 2007-08)

- 1- Décrire brièvement ce que l'on appelle des *propriétés d'équité*.
- 2- Une forme particulière de propriétés d'équité correspond à des énoncés de la forme : « *Si P est continuellement demandé, alors P sera obtenu (infiniment souvent)* ».

Pour cette forme de propriétés d'équité, on distingue entre *équité faible* et *équité forte*.

- a- Dans le cas de l'équité faible « *P est continuellement demandé* » est interprété par « *P est demandé sans interruption* ». En utilisant les propositions *demande_P* et *P*, modélisez ce type de propriété d'équité dans la logique temporelle linéaire propositionnelle (PLTL).
- b- Dans le cas de l'équité forte « *P est continuellement demandé* » est interprété par « *P est demandé de façon infiniment répétée (éventuellement avec des interruptions)* ». En utilisant les propositions *demande_P* et *P*, modélisez ce type de propriété d'équité dans la logique temporelle linéaire propositionnelle (PLTL).

- c- Est-ce que l'un de ces types de propriétés d'équité (forte ou faible) entraîne l'autre (justifiez) ?

Exercice 3 : logique temporelle CTL

Exprimer dans CTL*,

- 1- Tous les états satisfont p
- 2- On peut atteindre p par un chemin où q est toujours vraie
- 3- Quelque soit l'état on finit par revenir à l'état initial "init"
- 4- Quelque soit l'état, on peut revenir à l'état initial "init"

Exercice 4 : Structures de Kripke et PLTL

Soient 2 structures de Kripke S_1 et S_2 , et F une formule de PLTL. On notera $L(S_1)$ et $L(S_2)$, les langages respectivement reconnus par S_1 et S_2 . On suppose que $S_1 \models F$ une, que peut-on déduire si :

- a- $L(S_1) = L(S_2)$
- b- $L(S_2) \subseteq L(S_1)$
- c- $L(S_1) \subseteq L(S_2)$

Exercice 5 : Automates de Büchi

- 1- Ecrire un automate de Büchi sur l'alphabet $\{a,b\}$, reconnaissant le langage suivant : $b^*a(ba)^w$
- 2- Traduire les formules de PLTL suivantes en automates de Büchi sur l'alphabet $\{p, q, \neg p, \neg q\}$

Exercice 6 : équivalences de formules LTL

- 1- Exprimer le connecteur F à l'aide du connecteur U
- 2- Peut-on exprimer le connecteur de LTL X grâce à d'autres connecteurs ?

Exercice 7 : équivalences de formules CTL

Montrer que les connecteurs $\{true, \neg, \wedge, EX, E_U_ , A_U_ \}$, permettent de réexprimer tous les autres connecteurs de CTL..

Exercice 8 : sémantique CTL*

Définir la sémantique des nouveaux connecteurs suivants en se basant sur la définition de la sémantique des connecteurs de CTL* :

- a- $U_{\leq k}$ ("bounded until") : $p U_{\leq k} q$, p est vrai jusqu'à ce que q soit vrai et q est vrai dans au plus k observation.
- b- R ("release") : $p R q$, q est vraie jusqu'à (et inclus) le premier état où p est vraie sachant que p n'est pas forcément vraie un jour.

Exercice 9 : model checking de CTL

En se basant sur l'exemple donné dans le cours pour le connecteur EX, définir les procédures de vérification pour les connecteurs suivants $\{true, \neg, \wedge, E_U_ , A_U_ \}$.