

# TD 3 de Model Checking

---

## Model checking

**Exercice 1.** Jouez à la main l'algorithme de model checking de CTL (par marquage) sur le système de transition des slides pour la formule :  $\mathbf{AX}p \wedge p$

**Exercice 2.** Modifier l'algorithme pour gérer tous les cas suivants :  $\neg p$ ,  $\wedge$ ,  $\mathbf{AX}$ ,  $\mathbf{AG}$ ,  $\mathbf{AF}$ ,  $\mathbf{EF}$ ,  $\mathbf{AW}$ ,  $\mathbf{EW}$ ,  $\mathbf{AU}_{\leq k}$ ,  $\mathbf{EU}_{\leq k}$ ,  $\mathbf{AR}$ ,  $\mathbf{ER}$ .

**Exercice 3.** Transformez les propriétés de chemin suivantes en automates de Büchi sur alphabet  $\{p, \neg p\} \times \{q, \neg q\}$  (2 propriétés dans la structure de Kripke) :  $p$ ,  $\neg p$ ,  $\mathbf{X}p$ ,  $\mathbf{F}p$ ,  $\mathbf{G}p$ ,  $p\mathbf{U}q$ ,  $p\mathbf{W}q$ ,  $\mathbf{F}^\infty p$ ,  $\mathbf{G}^\infty p$ ,  $p\mathbf{U}_{\leq 3}q$ .

**Exercice 4.**

1. Exprimer en LTL les propriétés suivantes : (a) à l'instant suivant, si  $p$  vrai alors  $q$  n'est jamais vrai ; (b)  $p$  sera vrai au plus une fois ; (c)  $p$  sera vrai exactement 2 fois.
2. Pour chaque formule ci-dessus, dessinez l'automate de Büchi correspondant.

**Exercice 5.** La complémentation des automates de Büchi est très coûteuse. Proposer une manière de s'en passer.

**Exercice 6.** Comment tester le vide d'un automate de Büchi ? Comment calculer l'union et l'intersection d'automates de Büchi ?

**Exercice 7** (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.

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  $\varphi$  est la même formule, sans fair.

Enfin, donner une traduction de Fair CTL dans CTL\*.

**Exercice 8** (Preuve de MC CTL (\*)). Faire les preuves de correction et de complexité de l'algorithme de model checking de CTL.

**Exercice 9** (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  $\mathbf{A}\varphi_p$  ( $\varphi_p$  sans quantificateur) est vraie sur une structure de Kripke  $\mathcal{M}$  ssi tous les chemins de la structure satisfont la formule  $\varphi_p$ . 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  $\varphi_p$ .

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 propriété 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 10** (Automates de Büchi et  $LTL$  (\*)). à faire

**Exercice 11** (Automates de Büchi généralisés (\*)). à faire

**Exercice 12** (Automates de Büchi déterministes (\*)). à faire

**Exercice 13** (Automates de Büchi alternants (\*)). à faire

**Exercice 14** ( $CTL^*$  et automates d'arbres de Büchi (\*)). à faire

**Exercice 15** (entre  $CTL$  et  $CTL^*$  (\*)). à faire