

## Travaux dirigés numéro 2

Ceci est la fiche-TD<sup>1</sup> sur la logique temporelle (LTL, CTL, CTL\*). Dans le cas de certains exercices (surtout ceux de modélisation), vous en pourrez profiter pour découvrir NuSMV (commandes et validation de vos réponses lorsque cela est possible).

### Exercice 2.1 (Découverte de NuSMV)

Quand vous utilisez NuSMV, pensez à avoir toujours le manuel sous la main.

- NuSMV permet la description et la vérification de machine à états finis (FSM). Ecrivez un module `Feu.smv` décrivant un feu tricolore à la française.
- lancez NuSMV interactivement (option `-int`) en chargeant le fichier : `NuSMV -int Feu.smv`. Ceci peut aussi se faire en lançant NuSMV avec `NuSMV -int` puis en chargeant le modèle avec `read_model`.
- l'initialisation d'un système pour la vérification se fait avec la commande `go`.
- simulez le feu avec la commande `simulate -i` (vous devrez d'abord choisir un état initial avec `pick_state`).
- vérifiez les propriétés CTL suivantes à l'aide de la commande `check_spec -p` :  $EF \text{ couleur=rouge}$ ,  $AG \neg(\text{couleur=vert})$ ,  $A [ \neg(\text{couleur=vert}) \cup \text{couleur=vert} ]$ . Notez que lorsque la propriété est fautive, un contre-exemple est fourni. Expliquez comment utiliser l'outil pour fournir une trace-exemple d'une propriété.
- la vérification de formules LTL (réduites à CTL) se fait avec la commande `check_ltlspec`.
- sortez avec `quit`

### Exercice 2.2 (Formules LTL)

Dites si les formules suivantes sont bien formées ou non d'un point de vue LTL :

$G(r \wedge p)$ ,  $p \Rightarrow X(p \cup q)$ ,  $XX(q \wedge Gq)$ ,  $XU(p \vee Fq)$

### Exercice 2.3 (Formules CTL)

Dites si les formules suivantes sont bien formées ou non d'un point de vue CTL :

$EG r$ ,  $AG(q \Rightarrow EG r)$ ,  $A[p \cup EFr]$ ,  $E[A[p \cup q] \cup r]$ ,  $FG r$ ,  $XX r$ ,  $A \neg G \neg p$ ,  $AF[(r \cup q) \wedge (p \cup r)]$ ,  $(F r) \wedge (AG q)$

### Exercice 2.4 (Validité de formules LTL)

Prouver ou donner un contre-exemple de la validité des formules LTL suivantes :

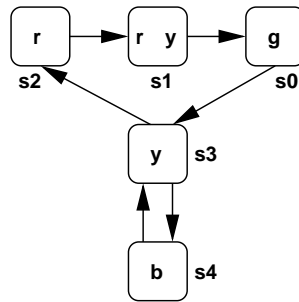
$Fp \Rightarrow Xp$ ,  $FGq \Rightarrow GFq$ ,  $(pUq) \wedge (pUr) \Rightarrow p \cup (q \wedge r)$ ,  $(GFp \wedge GFq) \Rightarrow (p \wedge q)$ .

### Exercice 2.5 (Vérification LTL)

---

<sup>1</sup>C'est-à-dire une fiche qui donne lieu à plusieurs TD.

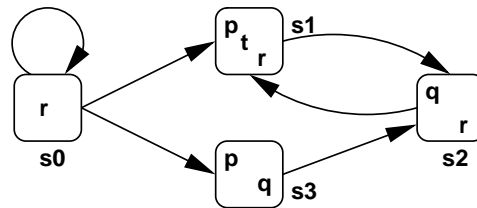
Soit l'automate  $\mathcal{M}$  suivant :



- vérifier pour quels états les formules LTL suivantes sont satisfaites :  $XXy$ ,  $XXr$ ,  $Fy$ ,  $GFy$ ,  $Fg$ ,  $\neg yUy$ ,  $\neg bUb$ .

### Exercice 2.6 (Vérification CTL)

Soit l'automate  $\mathcal{M}$  suivant :



- déplier l'automate  $\mathcal{M}$  à partir de  $s_0$  pour obtenir les chemins d'exécution de taille 4
- établir si les formules suivantes sont vérifiées pour  $\mathcal{M}, s_0$  :  $\neg p \Rightarrow r$ ,  $AF t$ ,  $\neg EG r$ ,  $E[t U q]$ ,  $EF q$ ,  $AG (r \vee q)$
- même question pour  $\mathcal{M}, s_2$

### Exercice 2.7 (Motifs CTL)

Les motifs les plus employés dans les spécifications CTL sont les suivants :

- $EF (\text{started} \wedge \neg \text{ready})$
- $AG (\text{requested} \Rightarrow AF \text{ acknowledged})$
- $AG p$
- $AG AF \text{ enabled}$
- $AF AG \text{ deadlock}$
- $AG \neg p \vee A[\neg p U q]$
- $AG EF \text{ restart}$
- $AG (\text{floor}=2 \wedge \text{direction}=\text{up} \wedge \text{buttonPressed5} \Rightarrow A[\text{direction}=\text{up} U \text{floor}=5])$

Donnez une intuition à chaque formule et pour chacune un exemple d'automates satisfaisant et ne satisfaisant pas la formule.

### Exercice 2.8 (Spécification en CTL)

Exprimer en CTL :

- à chaque fois que p est suivi par q en un nombre arbitraire d'étapes, alors r n'est pas vrai jusqu'à ce que t soit vrai
- après p, q n'est jamais vrai (sur tous les chemins)
- p précède s et t sur tous les chemins
- entre les évènements q et r, p n'est jamais vrai (sur tous les chemins)
- il existe au plus deux transitions qui mènent à un état satisfaisant q (sur tous les chemins)

### Exercice 2.9 (Equivalence de formules CTL)

Deux formules CTL  $\phi$  et  $\psi$  sont dites équivalentes, ce qui est noté  $\phi \equiv \psi$ , si et seulement si, pour tout modèle  $\mathcal{M}$ , pour tout état  $s$  de  $\mathcal{M}$ , on a :  $\mathcal{M}, s \models \psi \Leftrightarrow \mathcal{M}, s \models \phi$ .

Selon le cas, montrer l'équivalence ou exhiber un modèle qui invalide l'équivalence :

- $\neg AX\phi \equiv EX\neg\phi$
- $AG\phi \equiv \phi \wedge AXAG\phi$
- $AF\phi \vee AF\psi \equiv AF(\phi \vee \psi)$
- $EF\phi \equiv \phi \vee EXEF\phi$
- $AF\neg\phi \equiv \neg EG\phi$
- $A[\phi U \psi] \equiv \psi \vee (\phi \wedge AXA[\phi U \psi])$
- $A[\phi_1 UA[\phi_2 U \phi_3]] \equiv A[A[\phi_1 U \phi_2] U \phi_3]$

### Exercice 2.10 (Modélisation)

On désire modéliser un charriot élévateur. Ce charriot peut prendre deux positions, haute et basse. Si un bouton à l'étage est activé, alors l'élévateur va en position haute et y reste tant que le bouton est (ré-)activé. Dès que le bouton n'est plus activé, l'élévateur va en position basse.

Proposez un modèle qui permette de vérifier les propriétés suivantes (vous expliquerez ce qu'elles signifient et vous montrerez si elles sont vérifiées ou non en développant les algorithmes vus en cours) :  
AG ((on  $\wedge$  down)  $\Rightarrow$  AX  $\neg$  down), EG ((on  $\wedge$  down)  $\Rightarrow$  EX E[ $\neg$ down U  $\neg$ on])