

## TP LTL

NuSMV est installé sur les machines de la salle SAR. Vous pouvez également le télécharger et l'installer à partir du site : [nusmv.fbk.eu](http://nusmv.fbk.eu)

On nomme NUSMV le point d'installation ; le tutoriel et le manuel utilisateur sont dans le répertoire `($NUSMV)/share/nusmv/doc`. Un ensemble d'exemples de systèmes modélisés et analysés sont dans le répertoire `($NUSMV)/share/nusmv/examples`.

### Exercice 1

Construire les automates de Büchi correspondant aux formules suivantes, et vérifiez vos constructions à l'aide d'un outil de construction en ligne (LTL2BA, SPOT,...)

- (a)  $FG\ c$
- (b)  $GF\ c$
- (c)  $X\neg c \rightarrow XXX\ c$
- (d)  $G\ a$
- (e)  $a\ U\ G(b\ \vee\ c)$
- (f)  $(XXX\ b)\ U\ (b\ \vee\ c)$

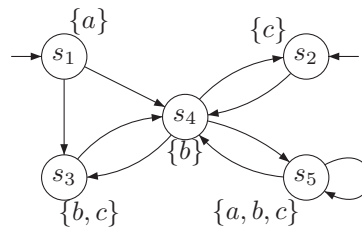


FIGURE 1 – Structure de Kripke pour l'exercice 2

### Exercice 2

- (a) Définir dans un fichier `exo1.smv` le modèle de la figure 2.
- (b) Simuler quelques exécutions dans NuSMV d'abord sans interaction, puis en mode interactif (regarder dans le tutorial Chapitre 3).
- (c) Vérifier à l'aide du model checker les propriétés suivantes :
  - (a)  $FG\ c$
  - (b)  $GF\ c$
  - (c)  $X\neg c \rightarrow XXX\ c$
  - (d)  $G\ a$
  - (e)  $a\ U\ G(b\ \vee\ c)$
  - (f)  $(XXX\ b)\ U\ (b\ \vee\ c)$

### Exercice 3

Voici un algorithme d'exclusion mutuelle pour 2 processus proposé par A. Pnueli. Chaque processus  $i$  ( $i = 0, 1$ ) possède une variable locale  $y_i$  valant initialement 0, et les deux processus partagent une variable  $s$  valant initialement 0. Le programme est présenté pour le processus  $i$ .

```
while true
Section non critique;
atom(y_i:=1, s:=i);
while (y_(1-i) !=0 and s==i)
wait;
Section critique;
y_i:=0;
```

- (a) Modéliser ce protocole en NuSMV
- (b) Simuler quelques exécutions
- (c) Exprimer la propriété d'exclusion mutuelle en LTL et la vérifier à l'aide du model checker
- (d) Vérifier l'absence de famine
- (e) Ajouter la contrainte d'équité **FAIRNESS running** à la spécification des processus dans le modèle et vérifier à nouveau l'absence de famine.
- (f) Exprimer en LTL le fait que chaque processus va entrer en section critique infiniment souvent. Vérifier cette propriété en utilisant la contrainte **FAIRNESS running**