

## TP1 - Systèmes de transitions en NuSMV

**Exercice 1:** Modéliser en NuSMV le système de transitions suivant :

Un lombric de 50 g, un millepatte de 30 g et une sauterelle de 20 g veulent passer la rivière. A leur disposition, une feuille d'arbre qui ne peut porter au maximum que 60 g.

---

**Exercice 2:** Modéliser en NuSMV le système de transitions correspondant à l'exécution en entrelacement des deux programmes suivants :

```
while (true) {           while (true) {
  flag1 := true ;        flag2 := true ;
  wait (!flag2)          wait (!flag1)
  section critique 1     section critique 2
  flag1 := false ;      flag2 := false ;
}                        }
```

Écrivez des formules LTL pour les propriétés suivantes, puis vérifiez-les sur le système de transitions NuSMV construit :

1. La variable *flag1* ne change de valeur que lorsque le premier programme exécute l'instruction 1 ou l'instruction 4.
2. Les deux programmes ne sont jamais en même temps en section critique.
3. Lorsque le programme 1 est à l'entrée de la section critique, il y entrera (dans un laps de temps quelconque).

Question subsidiaire : Comment expliquer le résultat donné par NuSMV lorsqu'il évalue cette formule ?

4. Le programme 1 se trouvera enfamé : il n'entrera jamais dans sa section critique.

Question subsidiaire : Comment expliquer l'apparente contradiction entre le résultat de l'évaluation de la formule écrite pour ce point et celle du point précédent ?

5. Le premier programme revient périodiquement à l'instruction 1.
- 

**Exercice 3:** L'algorithme du "boulangier" est une généralisation de la solution de Peterson au problème d'exclusion mutuelle pour  $n$  processus avec partage de mémoire. L'idée de l'algorithme est la suivante :

- La section critique signale, comme dans une boulangerie (ou rayon fromages, ou...) la conversation avec le vendeur pour acheter son pain (camembert, etc.).
- À son arrivée à la boulangerie chaque processus reçoit un ticket dont le numéro est supérieur aux numéros de tous les autres qui sont venus avant lui. (Attention à la finitude du domaine de valeurs des tickets ! vous devez prendre en compte le fait que le compteur des tickets soit "remis à zéro" lorsqu'il franchit sa valeur maximale !)
- Celui qui a le numéro le plus petit entre en section critique.

- Il se peut que plusieurs processus demandent en même temps un ticket et prennent le même ticket ! Lorsque cela se passe, celui dont le PID est le plus petit, dans l'ordre lexicographique, entre dans sa section critique.
- Après avoir fini sa section critique, et s'il a besoin d'y aller à nouveau, chaque processus prend un nouveau ticket.

Transformer cette spécification en pseudocode (avec pour 3 processus), modéliser le tout en NuSMV, puis proposer des formules LTL permettant de vérifier différentes propriétés du système : absence d'interblocage et exclusion mutuelle, mais aussi des formules formalisant la spécification donnée ci-dessus !

---

(Exercice supplémentaire)

**Exercice 4:** Modéliser en NuSMV la solution de M. Fisher au problème d'exclusion mutuelle :

prog\_i :

```
while true do
L : if id != 0 then goto L;
    id := i;
    pause(delay);
    if id != i then goto L;
    critical section;
    id := 0;
```

La correctitude de cette solution se base sur l'hypothèse que le temps d'exécution de chaque instruction est inférieur au `delay` utilisé à la quatrième ligne.

Modéliser cette solution en NuSMV, sous les hypothèses suivantes :

- Le système dans lequel les deux programmes s'exécutent possède *deux* processeurs – donc les deux programmes peuvent avancer *en même temps*.
- Chaque instruction peut prendre entre 1 et 2 unités de temps pour s'exécuter.
- Le `delay` est de 3 unités de temps.

1. *id* ne change de valeur que lorsque le programme 1 ou le programme 2 exécutent l'instruction 6 (le résultat devrait être évidemment faux !).
2. Les deux programmes ne sont jamais en même temps en section critique.
3. Lorsque le programme 1 est à l'entrée de la section critique, il y entrera (dans un laps de temps quelconque).

Question subsidiaire : Comment expliquer le résultat de l'évaluation de NuSMV de cette formule ?

4. Le programme 1 se trouvera enfamé : il n'entrera jamais dans sa section critique.

Question subsidiaire : Comment expliquer l'apparente contradiction entre le résultat de l'évaluation de la formule écrite pour ce point et celle du point précédent ?

5. Le premier programme revient périodiquement à l'instruction 1.

---