TD 2 — Vérification de propriétés avec spin

Exercice 1 — Modélisation et vérification d'un protocole d'élection

Un protocole d'élection vise à désigner un unique processus parmi un ensemble de processus. Le protocole que nous allons modéliser repose sur une topologie en anneau unidirectionnel, ce qui signifie que chaque processus recevra des messages uniquement depuis le processus précédent dans l'anneau et enverra des messages au processus suivant. Le principe du protocole est le suivant. Chaque processus sur l'anneau a un identifiant unique. C'est le processus qui a l'identifiant le plus élevé qui sera élu à l'issue du protocole. Les processus n'ont bien entendu aucune connaissance des identifiants des autres processus. Ils pourront découvrir ces identifiants en s'échangeant des messages.

Au lancement du protocole, chaque processus envoie au suivant un message contenant son identifiant. A la réception d'un identifiant supérieur au sien un processus sait alors qu'il ne peut pas être élu mais l'identifiant reçu est peut-être celui du futur élu. Il le retransmet alors au processus suivant. Par contre à la réception d'un identifiant inférieur au sien, un processus sait que l'identifiant reçu n'est pas celui du processus qui sera élu. Dans ce deuxième cas, il peut donc ignorer le message reçu. L'identifiant le plus grand de l'anneau est donc le seul qui fera le tour de l'anneau car il sera retransmis par chaque processus qui le recevra. Si un processus reçoit un message contenant son propre identifiant il sait donc qu'il est l'élu. Dans ce cas, il envoie un autre message au suivant indiquant qu'il a été élu. Le message fera le tour de l'anneau. Le but de cette deuxième phase du protocole est uniquement d'informer tous les processus de l'identité du processus élu. Le protocole se terminera à cet instant.

Q. 1.1 Ecrire un modèle promela de ce protocole à partir du squelette ci-dessous. À sa création, un processus recevra en argument le canal sur lequel il recevra des messages (depuis son prédécesseur dans l'anneau), le canal sur lequel il recevra des messages (vers son successeur dans l'anneau) ainsi que son identifiant. Le processus init déclarera N canaux de communications puis lancera N processus. À la fin du protocole, chaque processus devra afficher l'identité du processus élu.

```
proctype process(chan cin; chan cout; int id) { ... }
init {
    chan c1_2 = [1] of { ... };
    chan c2_3 = [1] of { ... };
    chan c3_1 = [1] of { ... };
    run process (c3_1, c1_2, 17);
    run process (c1_2, c2_3, 89);
    run process (c2_3, c3_1, 44);
}
```

Q. 1.2 Générer le graphe des états accessibles du modèle à l'aide de Spin.

Q. 1.3 Analyser les propriétés suivantes à l'aide de Spin :

1. Le protocole se termine toujours avec l'élection d'un processus.
2. Un seul processus est élu.
3. L'élection d'un processus est une propriété stable : si un processus est élu il le restera.
4. A la terminaison du protocole, tous les processus connaissent l'identité du processus élu.

Exercice 2 — Terminaison du crible d'Érathosthène

Q. 2.1 Modifier le programme promela du crible d'Ératosthène afin que le processus init envoie un message de terminaison aux processus de filtre. Ce message sera retransmis de proche en proche par les processus de la file. Le dernier processus de la file le transmettra ensuite au processus init.

Indication : utiliser un mtype pour différencier les messages (nombre à tester ou terminaison).

```
mtype = { NOMBRE, TERMINAISON };
chan c = [5] of { mtype, int };
```

Q. 2.2 Vérifier à l'aide de Spin que le programme termine dans tous les cas.