Théorie et Pratique de la Concurrence
TD Noté

Exercice 1 **(Propriétés)**: Pour chacune des phrases suivantes, indiquer quelle propriété elle exprime (sûreté ou vivacité) :

1. Si la machine à café est vide, elle sera rechargée ;
2. La machine à café se vide infiniment souvent ;
3. Si quelqu'un oublie un jeton dans la machine, il/elle ou quelqu'un d'autre utilisera le jeton pour s'acheter un café ;
4. Si la machine à café est vide, elle sera rechargée d'ici à une demi-heure ;
5. Un client ne quitte jamais le bar sans avoir consommé ;
6. Une fois qu'un client a pris sa consommation, il est sûr qu'il quittera le bar ;
7. Si la propriété A est satisfaite pour deux fois consecutives, alors elle sera satisfaite pour toujours ;
8. Si la propriété A est satisfaite pour deux fois consecutives, alors elle sera satisfaite jusqu'à ce qu'une autre propriété sera satisfaite à sa place.

Exercice 2 **(Logique Temporelle)**: On rajoute aux opérateurs vus en cours l'opérateur *suivant* $\bigcirc$. Formellement, soit s une suite infinie d'états $x_0x_1\ldots$ et soit A une formule LTL. $s, i \models \bigcirc A$ si et seulement si $s, i+1 \models A$.

Dire si les formules suivantes sont valides. Si oui prouvez-le, si non montrez un contrexemple.

1. $\Diamond A \iff A \vee \bigcirc \Diamond A$;
2. $\Box A \iff A \wedge (A \rightarrow \bigcirc A)$;
3. $\Box(A \rightarrow \bigcirc \neg A) \iff \Box(\bigcirc A \rightarrow \bigcirc \bigcirc \neg A)$;
4. $\Box(A \rightarrow \bigcirc \neg A) \iff \Box(\neg A \rightarrow \bigcirc A)$;
5. $\neg \Diamond A \iff \Box \neg A$;
6. $\Diamond(\neg A \wedge \bigcirc A) \iff \neg A \wedge \Diamond \bigcirc A$;
7. $\neg \bigcirc A \iff \bigcirc \neg A$;
8. $\Box A \iff A \wedge \bigcirc \Box A$;
9. $\neg \Box A \iff \Diamond \neg A$;
10. $\neg \Diamond \bigcirc A \iff \bigcirc \Diamond \neg A$;
11. $\Box(A \vee \bigcirc A) \iff \Box A \vee \Box \bigcirc A$.

Exercice 3 **(Exclusion Mutuelle)**: Choisir une des solutions vues en cours au problème

Exercice 4 **(Dîner des philosophes)**: Voici une implémentation du dîner des philosophes :

```
/* Les ressources = fourchettes */
#define NOMBRE_PHILO 5

#define FOURCHETTE_GAUCHE(p) ((p == 0) -> 4 : p-1)
#define FOURCHETTE_DROITE(p) ((p == 4) -> 0 : p+1)

int etat[NOMBRE_PHILO] = -1;           /* initialement tous pensent */

inline PENSER(p) { etat[p] = -1 }

inline AFAIM(p) { etat[p] = 0 }

inline MANGER(p) { etat[p] = 1 }

inline PRENDRE(p) { **** }

inline LIBERER(p) { **** }

/* Les philosophes */
active [NOMBRE_PHILO] proctype Philosophe() {
   int fourchette_gauche = FOURCHETTE_GAUCHE(_pid);   /* _pid et non pas _pid-1 */
   int fourchette_droite = FOURCHETTE_DROITE(_pid);   /* car il manque init     */

   do :: PENSER(_pid);
         AFAIM(_pid);
         PRENDRE(fourchette_droite);
         PRENDRE(fourchette_gauche);
         MANGER(_pid);
         LIBERER(fourchette_droite);
         LIBERER(fourchette_gauche);
   od
}
```

1. Implémenter les sémaphores.
2. Prouver par un contrexemple que cette solution ne satisfait pas l'absence de blocage.

Exercice 5 **(Lecteurs et écrivains)**: Considérez la solution suivante au problème des lecteurs et écrivains [Courtois et al., 1971], où on utilise des sémaphores binaires forts :

```
#define NOMBRE_ECRIVAINS 2
#define NOMBRE_LECTEURS 5

sémaphore ressource, mutex;

short readers;

active [NOMBRE_ECRIVAINS] proctype writer(short i) {
   do :: wait(ressource);

         /*          écriture          */

         signal(ressource);
   od
}

active [NOMBRE_LECTEURS] proctype reader(short i) {
   do :: wait(mutex);
         readers++;
         if :: (readers == 1) ->
            wait(ressource);
         fi;
         signal(mutex);

         /*          lecture           */

         wait(mutex);
         readers--;
         if :: (readers == 0) ->
            signal(ressource);
         fi;
         signal(mutex);
   od
}
```

1. Notez en particulier l'asymmétrie entre les deux types des processus :
   - Expliquer informellement pourquoi un lecteur peut lire sans avoir l'exclusivité sur la ressource ;
   - Montrez avec un contrexemple que, même sous hypothèse de progrès, un écrivain qui demande l'accès à la donnée peut être retardé infiniment.

BONUS Si l'on veut modifier la solution proposée pour qu'elle soit équitable, tout en gardant l'efficacité :
   - Quelles conditions minimales faut-il imposer pour qu'un lecteur puisse lire sans devoir attendre que la ressource se libère ?
   - Quelle autre variable globale faut-il ajouter ?
   - Faut-il un autre sémaphore pour la nouvelle variable ?

BONUS Et si on voulait obtenir l'alternance des lecteurs et écrivains, quelles modifications faut-il apporter à la solution proposée par Courtois et al. ?