

# Systemes concurrents

novembre 2019

1h30, documents autorisés : notes et supports distribués en cours, TD, TP.

Tous appareils électroniques prohibés.

Les exercices sont indépendants. Le barème est indicatif.

Lorsque le nombre de points n'est pas mentionné, la question compte pour un point.

Les notes seront (éventuellement) écrêtées à 20, qui est la note maximale.

*Conseil* : il est préférable de **lire attentivement** l'énoncé avant de répondre.

## 1 Transactions (4 points + 1 point de bonus)

### Questions

1. Le verrouillage à deux phases vous paraît-il un protocole de contrôle de concurrence adapté à la réalisation d'un service de mémoire transactionnelle? Justifiez votre réponse.
2. (**Bonus**) La mémoire transactionnelle s'appuie généralement sur une propagation en continu des mises à jour. Pourquoi est-ce que ce protocole est préféré à l'utilisation d'un journal après (espace de travail)?

Sur une plateforme disposant d'une mémoire transactionnelle, l'ellipse de verrous est une technique consistant à exécuter chaque section critique comme une transaction, et à répéter cette exécution jusqu'à ce que la transaction puisse valider, au lieu de protéger la section critique par l'acquisition et la libération d'un verrou d'exclusion mutuelle. Du point de vue de l'écriture des programmes, les deux approches ont la même simplicité : un mot clé (`synchronized`, `atomic...`) est associé à la section critique.

### Questions

3. L'ellipse de verrous garantit-elle le même niveau de cohérence des données partagées que l'usage de verrous d'exclusion mutuelle? Pourquoi?
4. L'ellipse de verrous est elle de nature à améliorer le degré de parallélisme? Pourquoi?
5. Est-il possible d'observer une situation de famine sur l'exécution d'une section critique lorsqu'on utilise l'ellipse de verrous? Justifiez votre réponse.

## 2 Mise en œuvre sur la plateforme Java (4 points)

Vous avez reçu le courriel suivant (les correspondants ont été rendus anonymes ; **le fichier joint** au courriel **se trouve sur la dernière page du sujet**)

De: A. F. <f.a@inp-toulouse.fr>  
À: M. N. <n.m@enseeiht.fr>  
Date: Wed, 22 Oct 2014 01:52:23 +0200  
Sujet: LecteurRedacteur  
Pièces jointes: 1 <LectRed\_PrioRedacteur.java>

Bonsoir monsieur M.,  
Est ce que vous voulez voir avec moi l'exercice de LecteurRedacteur,  
priorité aux rédacteurs.  
Quand je lance l'animation des fois ça marche, des fois non  
Merci pour votre assistance .  
Bonne soirée

**Question** Le programme joint comporte plusieurs erreurs, ainsi que des opérations superflues. Corrigez ce programme, afin qu'il réalise effectivement et proprement la stratégie « priorité aux rédacteurs », en commentant vos corrections, afin que votre correspondant puisse tirer profit de celles-ci.

Vous pouvez répondre directement sur le sujet (dans ce cas, **n'oubliez pas de mettre votre nom** sur celui-ci), ou sur votre copie, en faisant référence aux numéros de ligne du programme.

### 3 Machine chimique

On souhaite réaliser un moteur d'exécution pour un schéma de synchronisation inspiré des réactions chimiques. Une réaction chimique a lieu seulement lorsque tous les réactifs sont présents, et fournit des molécules produits. Par exemple, dans la réaction (R1) suivante :



Les molécules de dihydrogène et de dioxygène sont les réactifs, et les molécules d'eau sont le produit.

Ce phénomène va servir de base à un modèle de synchronisation de processus.

#### Données et structures

- une classe **Molécule**, permet de définir les différentes molécules disponibles. Une molécule
  - est identifiée par un attribut **formule** :  $\text{H}_2$  ou  $\text{O}_2$  dans l'exemple précédent.
  - possède une méthode **do()**, et une méthode **proceed()** destinées à être exécutées par les processus qui leur sont associés (cf infra).
- une classe **Réaction**, encapsulant une liste de molécules réactifs, (avec leurs coefficients supposés entiers) et une liste de molécules produits (avec leurs coefficients supposés entiers). Dans l'exemple précédent,  $\rho_1$  est une réaction, sa liste de réactifs est  $\langle\langle \text{H}_2, 2 \rangle; \text{O}_2, 1 \rangle\rangle$  et sa liste de produits est  $\langle\langle \text{H}_2\text{O}, 2 \rangle\rangle$
- une classe **Réacteur**, chargée de déclencher les différentes réactions lorsqu'elles sont possibles. Cette classe fournit
  - une méthode **rejoindre(m : Molécule)** qui permet d'introduire la molécule **m** parmi les molécules disponibles ,
  - et une méthode **ajouter(r : Reaction)**, qui permet d'ajouter la réaction **r** à la liste des réactions gérées par le réacteur.
- on suppose que les réactifs (molécules), les réactions, **ainsi qu'une instance R de la classe Réacteur** sont définies (et identifiées) à l'initialisation de l'application.

**Protocole d'usage** Le réacteur **R** va permettre de coordonner un ensemble de processus dont on suppose qu'ils respectent : les règles de comportement suivantes

- Chaque processus
  - est associé à sa création à une molécule, conservée dans un attribut **molecule** privé, qui lui est spécifique ;
  - exécute le code : `this.molecule.do()` ; `R.rejoindre(this.molecule)` ; `this.molecule.proceed()`

- un processus ayant appelé `R.rejoindre(m)` d'un réactif `m` donné reste en attente jusqu'à ce qu'il existe un appel à `R.rejoindre(ki)` (effectué par d'autres processus) pour chacun des autres réactifs `ki` nécessaires à l'une des réactions parmi les réactions gérées par le réacteur `R`.
- lorsque tous les appels à `R.rejoindre(-)` nécessaires à une réaction  $\rho$  sont effectués et en attente, ces appels se terminent de manière atomique, et le réacteur `R` poursuit en créant et lançant un processus pour chacune des molécules produit de la réaction  $\rho$ .

Dans l'exemple de la réaction ( $\rho_1$ ) précédente :

- Les molécules  $O_2$  et  $H_2$  sont définies, ainsi que la réaction  $\rho_1$
- Pour que la réaction  $\rho_1$  puisse être choisie par `R`, il faut que 2 processus associés à  $H_2$  soient créés et lancés, ainsi qu'un processus associé à  $O_2$ .
- Lorsque qu'un processus appelle `R.rejoindre(O2)`, il doit attendre (si ce n'est pas encore le cas) que deux autres processus appellent `R.rejoindre(H2)`
- Lorsque qu'un processus appelle `R.rejoindre(H2)`, il doit attendre (si ce n'est pas encore le cas) qu'un autre processus appelle `R.rejoindre(H2)` et qu'un autre processus appelle `R.rejoindre(O2)`
- Lorsque 2 processus ont appelé `R.rejoindre(H2)` et 1 processus a appelé `R.rejoindre(O2)`, ces processus sont libérés, et 2 processus associés à  $H_2O$  sont créés et lancés par le réacteur.

### 3.1 Culture (1 point)

6. Le schéma de synchronisation présenté est proche d'un schéma présenté en cours. Lequel? Quelles sont les différences (est-ce une variante, une généralisation, une spécialisation)?

### 3.2 Synthèse de l'eau (8 points)

**Synchronisation par sémaphores** On souhaite implémenter exactement et uniquement le schéma correspondant à la réaction  $\rho_1$ , à l'aide de sémaphores. En d'autres termes, on recherche une solution pour le cas précis où seules ( $\rho_1$ ),  $O_2$  et  $H_2$  sont définies. Le code de la méthode `rejoindre(-)`, appelé par les différents processus pourra alors avoir la structure :

```
rejoindre(m : Molécule) {
    si (m=O2) alors traiter_O2();
    sinon traiter_H2();
    finsi;
    lancer(H2O); /* crée et lance un nouveau processus associé à la molécule H2O
    lancer(H2O); * à condition que le nombre de processus déjà lancés
                  * associés à la molécule H2O soit ≤ 1
                  */
}
```

7. (3 points) On suppose que l'on dispose de sémaphores, définis par une classe `Semaphore`, fournissant les opérations `P()` et `V()`.

Donner une implémentation pour les opérations `traiter_O2()` et `traiter_H2()`, réalisant le schéma de synchronisation voulu. Il est demandé de fournir une solution directe au problème, utilisant les sémaphores pour transmettre des événements et/ou limiter le degré de parallélisme des processus, selon les schémas illustrés par les premières questions du problème du barbier ou de celui des philosophes vus en TD.

**Synchronisation par moniteurs** Dans le même contexte que la question précédente (seules  $\rho_1$ ,  $O_2$  et  $H_2$  sont définies), on souhaite définir un moniteur pour gérer la progression des processus synchronisés par réactions. Ce moniteur correspondra donc au réacteur  $R$ , et son interface du moniteur se réduit à l'opération `rejoindre(m : Molécule)`.

On suppose que les moniteurs disponibles sont des **moniteurs de Hoare** (exécution en exclusion mutuelle des opérations implicite/automatique, priorité au signalé, files FIFO associées aux variables condition)

8. Il existe beaucoup de similitudes entre les appels à `attendre()/signaler()` dans le cas des moniteurs et les appels à `P()/V()` dans le cas des sémaphores. Peut-on s'appuyer sur ces similitudes pour transposer simplement dans le cadre des moniteurs la solution directe développée avec les sémaphores ? Si oui, comment ? Si non, pourquoi ?
9. On se propose de réaliser le moniteur en suivant la démarche « standard » vue en TD, basée sur l'évaluation de conditions d'état pour contrôler la progression des processus.
  - (a) Ecrire (en français) la condition d'acceptation de l'opération `rejoindre(-)`.
  - (b) Définir les variables d'état permettant de représenter cette condition.
  - (c) (2 points) Programmer l'opération du moniteur. Préciser (sous forme de commentaire dans le code) les pré/post conditions des `signaler/attendre`.

### 3.3 Réacteur général (3 points + 8 points de bonus)

On souhaite maintenant traiter le cas, général, où l'on dispose d'un ensemble de molécules, pouvant intervenir dans un ensemble de réactions. Pour faciliter la programmation des solutions, on pourra considérer que les identifiants de molécules et les identifiants de réactions sont des types énumérés (autrement dit qu'ils peuvent être utilisés comme indices de tableaux). Il est important de remarquer qu'une même molécule peut intervenir dans plusieurs réactions, mais ne pourra être effectivement utilisée que dans une seule, à un moment donné. Par exemple, si l'on dispose initialement de molécules S,  $O_2$ ,  $H_2$ , et  $H_2S$  on peut envisager (en particulier) les réactions :

- ( $\rho_1$ )  $2 H_2 + O_2 \rightarrow 2 H_2O$
- ( $\rho_2$ )  $S + O_2 \rightarrow SO_2$
- ( $\rho_3$ )  $2 H_2S + 3 O_2 \rightarrow 2 H_2O + 2 SO_2$

Un appel à `R.rejoindre(O2)` pourra donc être utilisé pour l'une des réactions  $\rho_1$ ,  $\rho_2$ , ou  $\rho_3$  en fonction de l'arrivée des autres réactifs, et éventuellement en fonction de critères de choix spécifiques, lorsque plusieurs choix sont simultanément possibles.

**Synchronisation par moniteurs** Il s'agit de construire le moniteur correspondant au cas général, en utilisant les moniteurs de Hoare et la démarche « standard ». L'interface de ce moniteur fournit l'opération `rejoindre(-)`

10. Ecrire (en français) la condition d'acceptation de l'opération `rejoindre(-)`
11. Définir les variables d'état permettant de représenter cette condition.
12. (**Bonus**, 4 points) Programmer l'opération du moniteur. Préciser (sous forme de commentaire dans le code) les pré/post conditions des `signaler/attendre`.
13. Si pour chacun des réactifs  $m$ , l'opération `rejoindre(m)` se trouve appelée infiniment souvent, la solution construite est-elle équitable ? (C'est-à-dire : tout processus est-il assuré de terminer son opération `rejoindre(m)` au bout d'un temps fini ?). Justifiez votre réponse. Note : il n'est pas demandé d'écrire une solution équitable. Il est simplement demandé d'évaluer l'équité de la solution proposée.

**Tâches Ada (Bonus, 2 points)**

14. (2 points) Ecrire une tâche serveur Ada réalisant le même service que le moniteur défini dans les questions 10) à 12)

**Synchronisation par sémaphores (Bonus, 2 points)**

15. La solution développée dans la question 7) a l'avantage d'être efficace, car exempte de conflits d'accès à des variables partagées, mais elle a l'inconvénient d'être ad-hoc, et difficile à généraliser.
- (a) Quelle est la principale difficulté qui se présente avec l'approche choisie initialement, si l'on cherche à traiter le cas général, où l'on doit considérer un ensemble de réactions, mettant en jeu un ensemble de molécules ?
  - (b) Donnez les principes de réalisation d'une solution simple à base de sémaphores, dans le cas général. Note : il n'est pas demandé d'écrire une solution, mais de donner un schéma de solution.