

Epreuve Écrite



La clarté de la rédaction et la justification des réponses sont des éléments essentiels de l'appréciation. Les exercices sont indépendants. Le nombre d'exercices est quatre (4) qui sont tous à faire.

Ecrit

On rappelle pour les trois prochains exercices qu'un programme PlusCal peut contenir plusieurs processus et que chaque processus est traduit par une relation next spécifique. Le système global est alors défini par la disjonction des nexts composant le programme. Ensuite, on traduit automatiquement le programme et ses processus en une suite d'actions et de relations next. Une des relations next est utilisée pour le modèle qui produit toutes les exécutions possibles selon la relation next à partir d'une assertion définissant les conditions initiales. Vous avez utilisé l'outil PlusCal et vous avez testé des programmes avec plusieurs processus partageant des variables communes. Une instruction particulière est l'instruction assert A qui revient à tester à l'exécution si les valeurs courantes des variables satisfont A. Comme l'outil engendre toutes les exécutions possibles, si pour une exécution A est faux, l'outil stoppera et vous informera de cet état. Dans les exercices 1, 2 et 3 vous devez trouver une condition pour que le assert ne soit pas faux pour une exécution possible. Pour cela, vous devez simuler à la main les différents cas d'exécution et proposer une assertion qui convienne. Evidemment si vous utilisez TRUE, cela va fonctionner mais cette réponse n'est pas celle attendue et votre assertion doit contenir au moins une occurrence de variables du programme.

Exercice 1. On considère un système formé de deux processus one et two assurant les calculs suivants:

- one: le processus envoie les entiers pairs entre 0 et N via un canal de communication à two.
- two: le processus reçoit les valeurs envoyées par one et ajoute la valeur reçue à la variable s.
- three: le processus fait un calcul de la somme des entiers de 0 à N/2.

On suppose que N est divisible par quatre (4) et positif non nul.

Question 1.1. Montrer par récurrence que $\sum_{i=0}^n 4 * i = 2 * n * (n + 1)$

Question 1.2. Afin de vérifier que le calcul effectué par les deux processus est correct, on décide de vérifier que, quand tous les processus ont terminé la variable result contient la somme des entiers multiples de 4 entre 0 et N. Expliquer comment on peut vérifier cette propriété avec l'outil ToolBoxTLA en utilisant la propriété de la question précédente.

Question 1.3. Plutôt que d'utiliser la propriété de la question 1, on décide de calculer avec le processus three la somme des entiers de 0 à N%4. Proposer une propriété à vérifier afin de montrer que le calcul du processus two est correct.