

TP 8 Systèmes concurrents et distribués

Exercice 1 : Complétez la preuve du résultat vu au cours que le problème du consensus ne peut être résolu avec un algorithme wait-free pour trois threads en utilisant des instructions test-and-set. Il faut discuter tous les cas possibles. Dans la preuve vue au cours on a seulement discuté ce qui se passe si P0 et P1 exécutent une opération TestAndSet sur la même variable (partagée) `tas`. Pour vous aider, commencer par lister quelles sont les opérations atomiques que peuvent exécuter P0 et P1.

Exercice 1 : Réécrivez la preuve de l'algorithme pour résoudre le problème du consensus pour n threads avec l'instruction `move`. Utilisez les relations de précédences (flèches) comme formalisme. La preuve se trouve aux pages 290-291.

Vous devez montrer :

1. P0 doit être supérieur à j1 (P0 est le thread P0 et c'est aussi son identificateur qui peut être 0, 1, ..., n-1.)
2. On montre qu'il existe une suite de thread P2, P3, etc. dont les identificateurs satisfont $P0 > P2 > P3 > \dots$ et $P_i > j1$, que les threads exécutent le `move(r[Pi,0],r[Pi,1])` avant le `move(r[P0,0],r[P0,1])` et que le résultat du `move` est `r[Pi,1]=Pi-1` (la valeur n'est pas choisie)
3. Concluez qu'il y a une contradiction.

