

BLIN Sébastien
COLLIN Pierre-Henri



École supérieure d'ingénieurs de Rennes

1ère Année
Parcours Informatique

Algorithmie des graphes

Dossier de Spécifications - Allocateur

Sous l'encadrement de :
Bateux Quentin

1 Règles et définitions supplémentaires

- **R12** : En cas d'interblocage, le processus ayant créé la demande menant à cet interblocage est détruit.
- **R13** : Un processus actif peut demander de nouvelles ressources même si ça mène à un interblocage.
- **R14** : Un processus qui ne possède pas de ressources est actif (car toutes ses demandes sont satisfaites).

2 Règles de modélisation

2.1 Invariants

- **MI1** : Les sommets des graphes représentent des processus et des ressources.
- **MI2** : Les sommets de sorties du graphe représentent les ressources.
- **MI3** : Les sommets d'entrées du graphe représentent les processus actifs.
- **MI4** : Tout sommet qui n'est ni d'entrée, ni de sortie est un processus bloqué.
- **MI5** : Une demande de ressource est représenté par un arc orienté entre un sommet-processus vers un sommet-ressource.
- **MI6** : Une attente entre processus est représenté par un arc orienté du sommet-processus s'exécutant le premier vers un sommet-processus qui s'exécutera plus tard.
- **MI7** : Une file d'attente d'une ressource est représenté par le chemin le plus long entre le sommet-processus actif de la ressource et celle-ci.
- **MI8** : Un interblocage est modélisé par un cycle dans le graphe. Ce cycle est détecté à l'aide de l'algorithme de Tarjan qui montre les composantes fortement connexes.
- **MI9** : L'affichage décrit par (O5) est donné par la file d'attente de la ressource (sans les processus actifs).
- **MI10** : L'affichage des attentes d'un processus est donné par les parents de celui-ci.

2.2 Variants

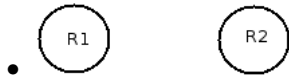
- **MV1** : La création d'un processus entraîne la création d'un sommet-processus.
- **MV2** : La destruction d'un processus actif entraîne la destruction de tous ces arcs.
- **MV3** : Si une demande provoque un interblocage, on détruit le sommet-processus venant de réaliser la demande.
- **MV4** : La demande d'une ressource libre par un processus entraîne la création d'un arc entre le sommet-processus et le sommet-ressource.
- **MV5** : La demande d'une ressource non-libre par un processus entraîne la création d'un arc entre le sommet-processus et le sommet-ressource ainsi qu'un arc orienté du dernier sommet-processus en attente par ressource demandée vers le sommet.
- **MV6** : La libération d'une ressource entraîne la suppression de l'arc entre le sommet-processus et le sommet-ressource et entraîne la redirection de ses arcs d'entrées vers ses fils en prenant en compte les fils d'attentes.

3 Scénarios

Nous considérons un allocateur disposant des ressources R1,R2 :

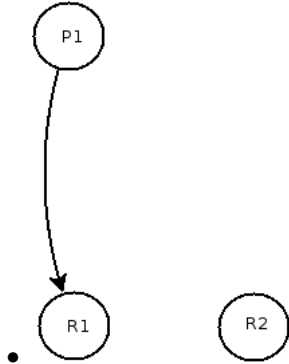
3.1 Acte 1

- ordre = création d'un processus P1 (O1)
- graphe = P1, R1, R2
- règles = R2, MV1



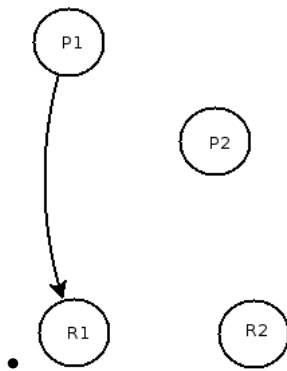
3.2 Acte 2

- ordre = demande de ressources R1 par le processus P1 (O3)
- graphe = P1, R1, R2
- file R1 = P1
- règles = R3, R6, MV4



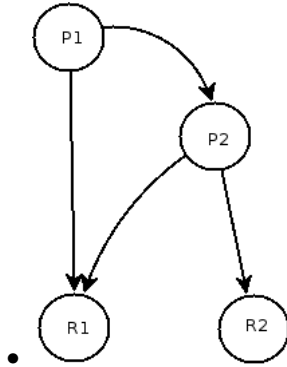
3.3 Acte 3

- ordre = création d'un processus P2 (O1)
- graphe = P1, P2, R1, R2
- file R1 = P1
- règles = R2, MV1



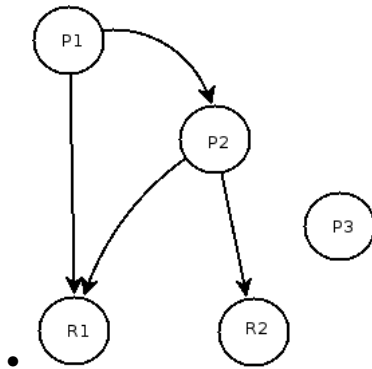
3.4 Acte 4

- ordre = demande de ressources R1 et R2 par le processus P2
- graphe = P1, P2, R1, R2
- file R1 = P1, P2
- file R2 = P2
- règles = R1, R3, R4, R7, R8, MV5



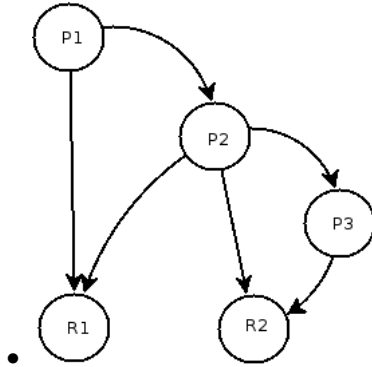
3.5 Acte 5

- ordre = création d'un processus P3
- graphe = P1, P2, P3, R1, R2
- file R1 = P1, P2
- file R2 = P2
- règles = R2, R7, MV1



3.6 Acte 6

- ordre = demande de ressources R2 par le processus P3
- graphe = P1, P2, P3, R1, R2
- file R1 = P1, P2
- file R2 = P2, P3
- règles = R1, R4, R7, MV5

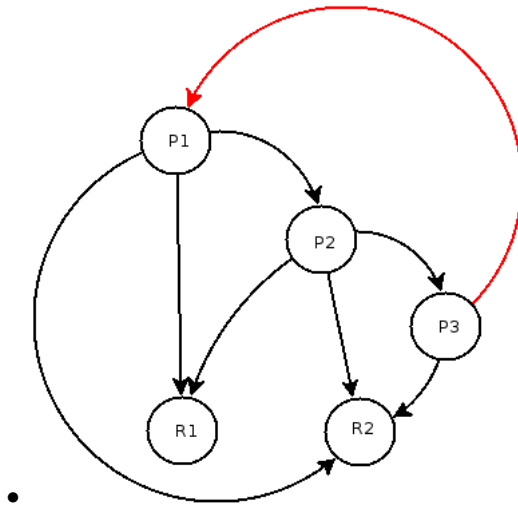


3.7 Acte 7

- ordre = affichage des fils d'attente par ressources
- graphe = P1, P2, P3, R1, R2
- file R1 = P1, P2
- file R2 = P2, P3, P1
- règles = R7, MI9

3.8 Acte 8

- ordre = demande de ressources R2 par le processus P1
- graphe = P1, P2, P3, R1, R2
- file R1 = P1, P2
- file R2 = P2, P3, P1
- règles = R1, R4, R7, R13, MV5

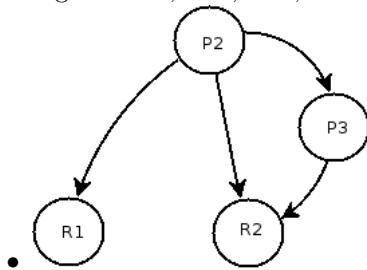


3.9 Acte 9

- ordre = affichage des processus concernés par un interblocage
- graphe = P1, P2, P3, R1, R2
- file R1 = P1, P2
- file R2 = P2, P3, P1
- règles = R7, MI8

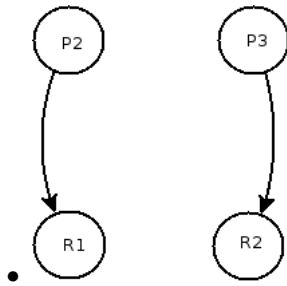
3.10 Acte 10

- ordre = destruction du processus P1
- graphe = P2, P3, R1, R2
- file R1 = P2
- file R2 = P2, P3
- règles = R7, R11, R12, MV2, MV3



3.11 Acte 11

- ordre = libération de la ressource R2 par le processus P2
- graphe = P2, P3, R1, R2
- file R1 = P2
- file R2 = P3
- règles = R5, R7, R9, MV5, MV6

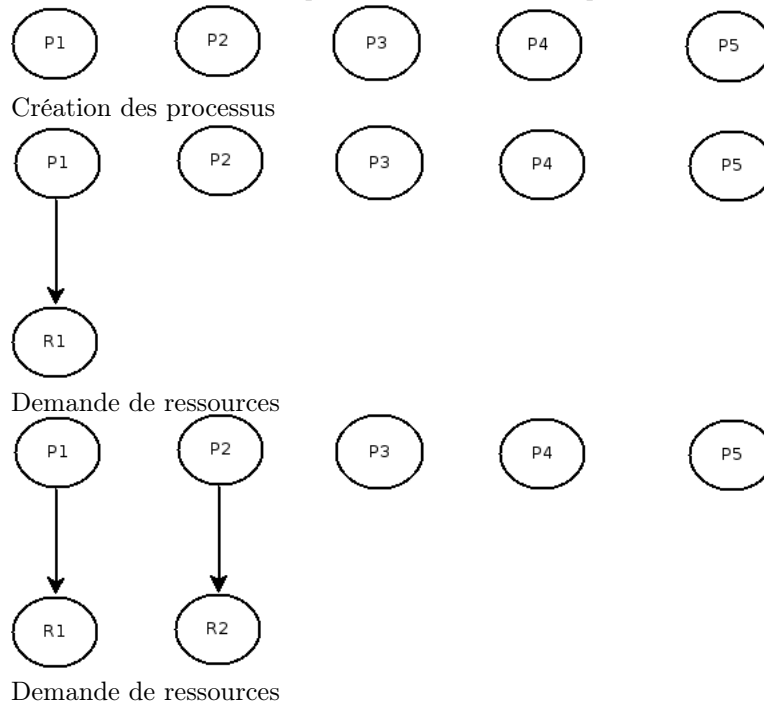


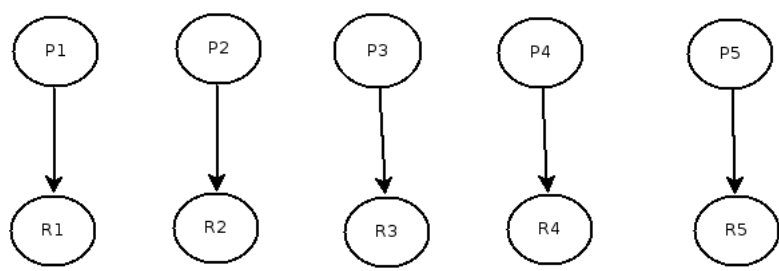
3.12 Acte 12

- ordre = affichage des processsus actifs
- graphe = P2, P3, R1, R2
- file R1 = P2
- file R2 = P3
- règles = R6, MI3

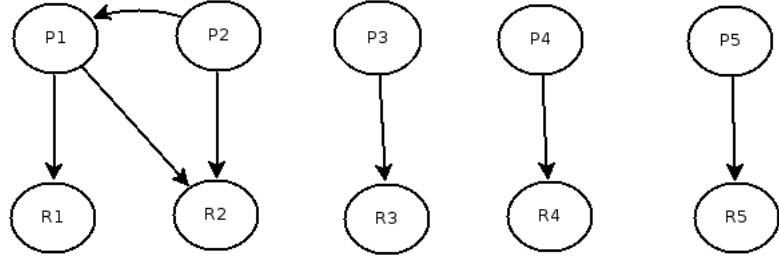
4 Tests

Tous les tests ont été passé avec succès. Exemple avec le test 5 (dans bin/tests/test5.entree)

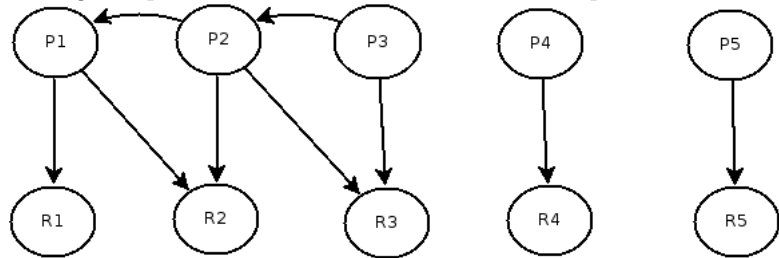




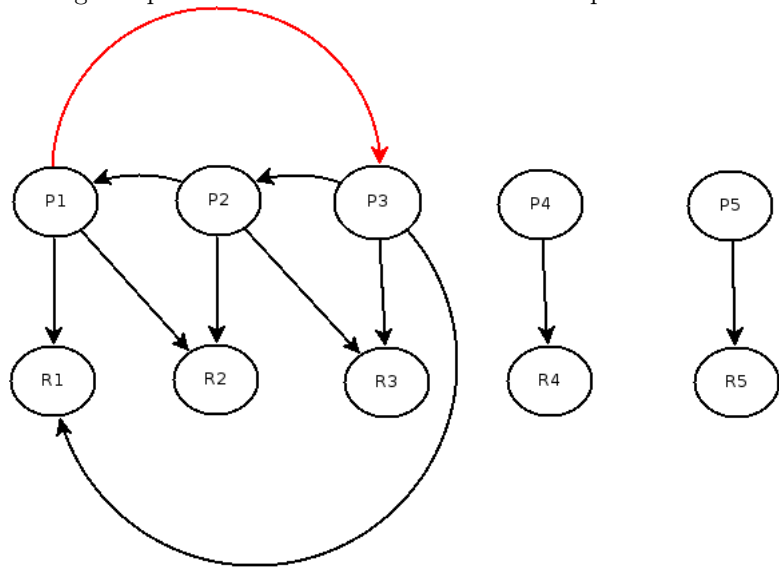
Demande de ressources



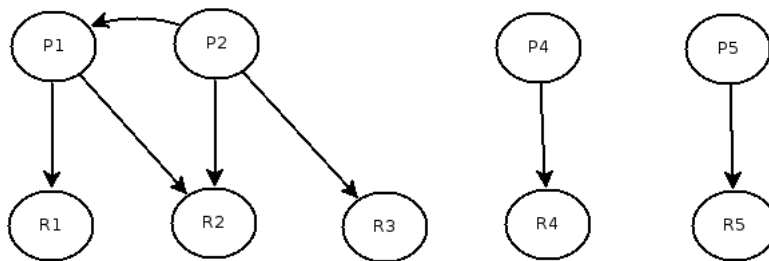
Blocage du processus 1 car la ressource 2 est occupée



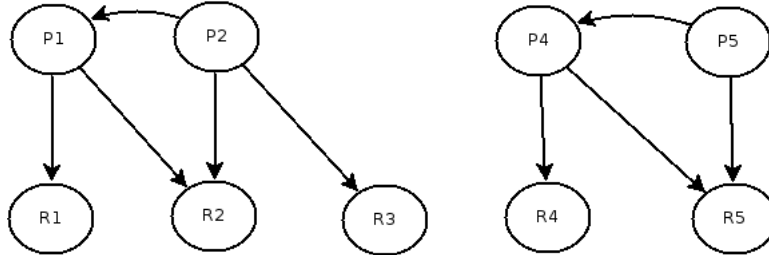
Blocage du processus 2 car la ressource 3 est occupée



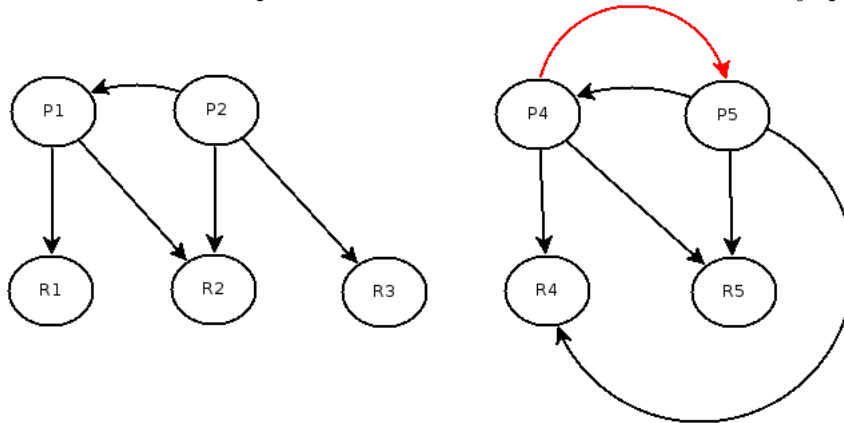
Interblocage



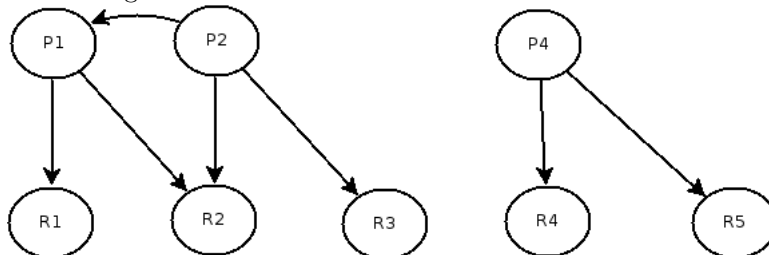
Destruction du processus 3 causant l'interblocage



Mise en attente d'un processus 4 car la ressource 5 demandée est déjà prise



Interblocage



Destruction du processus 5 causant l'interblocage