

TD Transaction

Exercice 1

- 1) Décrire les ordonnancements non-série que l'on peut constituer avec T_i et T_j ? Parmi ces ordonnancements, quels sont ceux qui sont sérialisables (justifier)?
- $T_i \ T_j: R_i(A) \ W_i(B) \ R_j(A) \ R_j(B)$ est un ordonnancement en série
- $T_j \ T_i: R_j(A) \ R_j(B) \ R_i(A) \ W_i(B)$ est un ordonnancement en série

Exercice 1

- Dans un ordonnancement non série, les transactions sont exécutées en parallèle.
- Pour construire les ordonnancements non série, on va permuter les actions de l'ordonnancement en série. Cependant, l'ordre des actions de chaque transaction doit être respecté.
- Par exemple, dans l'ordonnancement non série, pour T_i , $R_i(A)$ doit apparaître avant $W_i(B)$ et pour T_j , $R_j(A)$ doit apparaître avant $R_j(B)$.
- On ne peut pas alors avoir: **Wi(B) Ri(A) Rj(B) Rj(A) (faux)**
- Ou encore: **Ri(A) Rj(B) Rj(A) Wi(B) (faux)**

Exercice 1

- Les ordonnancement non série possibles sont:
- $T_i T_j: R_i(A) \textcolor{red}{W_i(B)} R_j(A) R_j(B)$ (en série)
 - 1) $R_i(A) R_j(A) R_j(B) \textcolor{red}{W_i(B)}$
 - 2) $R_i(A) R_j(A) \textcolor{red}{W_i(B)} R_j(B)$
- $T_j T_i: R_j(A) \textcolor{red}{R_j(B)} R_i(A) W_i(B)$ (en série)
 - 3) $R_j(A) R_i(A) W_i(B) \textcolor{red}{R_j(B)}$
 - 4) $R_j(A) R_i(A) \textcolor{red}{R_j(B)} W_i(B)$

Exercice 1

- Pour vérifier si un ordonnancement est sérialisable, il faut dessiner son graphe de précédence ($wi-wj$, $wi-Rj$, $Ri-wj$):
- Si Ti écrit ($wi(A)$) ensuite Tj écrit ($wj(A)$) sur le même granule alors Ti précède Tj



- Si Ti écrit ($wi(A)$) ensuite Tj lit ($Rj(A)$) sur le même granule alors Ti précède Tj



- Si Ti lit ($Ri(A)$) ensuite Tj écrit ($wj(A)$) sur le même granule alors Ti précède Tj



Exercice 1

- 1) $R_i(A)$ $R_j(A)$ **$R_j(B)$** $W_i(B)$
- Conflit sur A: pas de conflit
- Conflit sur B: $R_j(B)$ $W_i(B)$ donc T_j précède T_i



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

- 2) $R_i(A)$ $R_j(A)$ $W_i(B)$ $R_j(B)$
- Conflit sur A: pas de conflit
- Conflit sur B: $W_i(B)$ $R_j(B)$ donc T_i précède T_j



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

- 3) $Rj(A)$ $Ri(A)$ $Wi(B)$ $Rj(B)$
- Conflit sur A: pas de conflit
- Conflit sur B: $Wi(B)$ $Rj(B)$ donc Ti précède Tj



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

- 4) Rj(A) Ri(A) **Rj(B) Wi(B)**
- Conflit sur A: pas de conflit
- Conflit sur B: Rj(B) Wi(B) donc Tj précède Ti



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

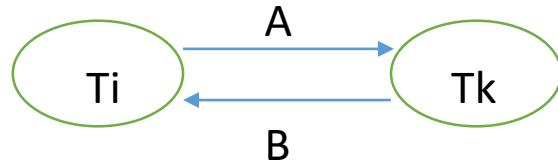
- 2) Décrire les ordonnancements non-série que l'on peut constituer avec Ti et Tk ? Parmi ces ordonnancements, quels sont ceux qui sont serialisables (justifier) ?
- TiTk: Ri(A) Wi(B) wk(A) wk(B) (en série)
- TkTi: wk(A) wk(B) Ri(A) Wi(B) (en série)
- Dans les ordonnancement non série Ri(A) doit être avant Wi(B) et wk(A) avant wk(B)

Exercice 1

- Les ordonnancement non série possibles sont:
- $T_i T_k: R_i(A) \text{ Wi(B)} W_k(A) W_k(B)$ (en série)
 - 1) $R_i(A) W_k(A) W_k(B) \text{ Wi(B)}$
 - 2) $R_i(A) W_k(A) \text{ Wi(B)} W_k(B)$
- $T_k T_i: W_k(A) \text{ Wk(B)} R_i(A) W_i(B)$ (en série)
 - 3) $W_k(A) R_i(A) W_i(B) \text{ Wk(B)}$
 - 4) $W_k(A) R_i(A) \text{ Wk(B)} W_i(B)$

Exercice 1

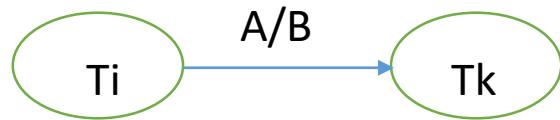
- 1) $Ri(A)$ $Wk(A)$ $Wk(B)$ $Wi(B)$
- Conflit sur A: $Ri(A)$ $Wk(A)$ donc Ti précède Tk
- Conflit sur B: $Wk(B)$ $Wi(B)$ donc Tk précède Ti



- Il y a un cycle dans ce graphe, cet ordonnancement est alors **non sérialisable**.

Exercice 1

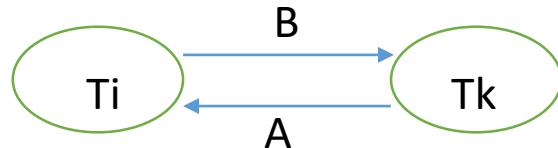
- 2) $Ri(A)$ $Wk(A)$ $Wi(B)$ $Wk(B)$
- Conflit sur A: $Ri(A)$ $Wk(A)$ donc Ti précède Tk
- Conflit sur B: $Wi(B)$ $Wk(B)$ donc Ti précède Tk



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

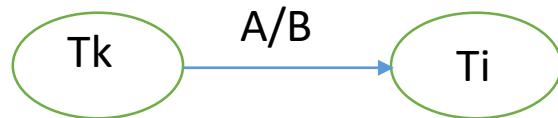
- 3) $Wk(A)$ $Ri(A)$ $Wi(B)$ $Wk(B)$
- Conflit sur A: $Wk(A)$ $Ri(A)$ donc Tk précède Ti
- Conflit sur B: $Wi(B)$ $Wk(B)$ donc Ti précède Tk



- Il y a un cycle dans ce graphe, cet ordonnancement est alors **non sérialisable**.

Exercice 1

- 4) $Wk(A)$ $Ri(A)$ $Wk(B)$ $Wi(B)$
- Conflit sur A: $Wk(A)$ $Ri(A)$ donc Tk précède Ti
- Conflit sur B: $Wk(B)$ $Wi(B)$ donc Tk précède Ti



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

- Principe de sérialisabilité: ne laisser s'exécuter les transactions en parallèle que celles provoquant les mêmes effets sur les données qu'une exécution en série (en séquence) de ces mêmes transactions.
- Par exemple dans cet ordonnancement non série et non sérialisable:
1) **Ri(A)** **Wk(A)** **Wk(B)** **Wi(B)**
- Dans l'ordonnancement en série on doit normalement avoir en sortie la nouvelle valeur écrite par Tk: **Ri(A)** **Wi(B)** **Wk(A)** **Wk(B)**, alors que l'ordonnancement 1) **Ri(A)** **Wk(A)** **Wk(B)** **Wi(B)** a en sortie pour B la valeur écrite par Ti, cet ordonnancement pour la donnée B ne donne pas le même effet que l'ordonnancement en séquence (en série).

Exercice 1

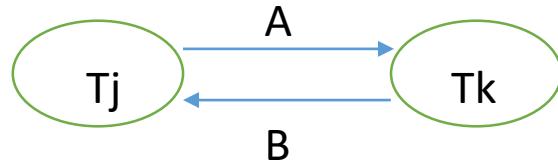
- 3) Décrire les ordonnancements non-série que l'on peut constituer avec T_j , T_k ? Parmi ces ordonnancements, quels sont ceux qui sont sérialisables (justifier)?
- T_jT_k : $R_j(A)$ $R_j(B)$ $W_k(A)$ $W_k(B)$ (en série)
- T_kT_j : $W_k(A)$ $W_k(B)$ $R_j(A)$ $R_j(B)$ (en série)
- Dans les ordonnancement non série $R_j(A)$ doit être avant $R_j(B)$ et $W_k(A)$ avant $W_k(B)$

Exercice 1

- Les ordonnancement non série possibles sont:
- TjTk: Rj(A) **Rj(B)** Wk(A) Wk(B) (en série)
- 1) Rj(A) Wk(A) Wk(B) **Rj(B)**
- 2) Rj(A) Wk(A) **Rj(B)** Wk(B)
- TkTj: Wk(A) **Wk(B)** Rj(A) Rj(B) (en série)
- 3) Wk(A) Rj(A) Rj(B) **Wk(B)**
- 4) Wk(A) Rj(A) **Wk(B)** Rj(B)

Exercice 1

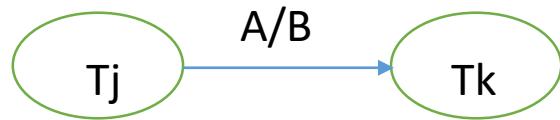
- 1) 1) $Rj(A)$ $Wk(A)$ $Wk(B)$ $Rj(B)$
- Conflit sur A: $Rj(A)$ $Wk(A)$ donc Tj précède Tk
- Conflit sur B: $Wk(B)$ $Rj(B)$ donc Tk précède Ti



- Il y a un cycle dans ce graphe, cet ordonnancement est alors **non sérialisable**.

Exercice 1

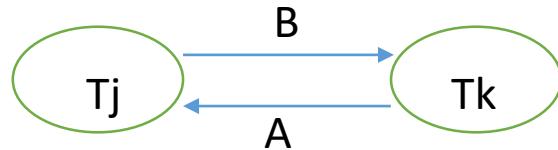
- 2) $Rj(A)$ $Wk(A)$ $Rj(B)$ $Wk(B)$
- Conflit sur A: $Rj(A)$ $Wk(A)$ donc Tj précède Tk
- Conflit sur B: $Rj(B)$ $Wk(B)$ donc Tj précède Tk



- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

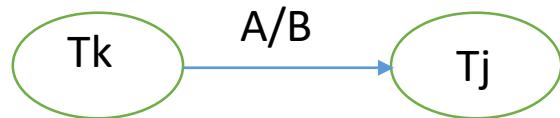
- 3) $Wk(A)$ $Rj(A)$ $Rj(B)$ $Wk(B)$
- Conflit sur A: $Wk(A)$ $Rj(A)$ donc Tk précède Tj
- Conflit sur B: $Rj(B)$ $Wk(B)$ donc Tj précède Tk



- Il y a un cycle dans ce graphe, cet ordonnancement est alors **non sérialisable**.

Exercice 1

- 4) $W_k(A)$ $R_j(A)$ $W_k(B)$ $W_j(B)$
- Conflit sur A: $W_k(A)$ $R_j(A)$ donc T_k précède T_j
- Conflit sur B: $W_k(B)$ $W_j(B)$ donc T_k précède T_j



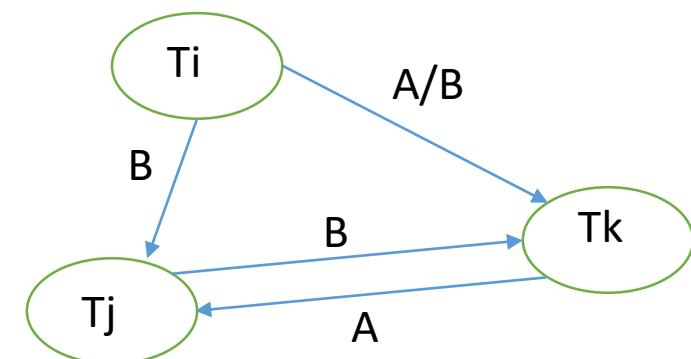
- Il n'y a pas de cycle dans ce graphe, cet ordonnancement est alors sérialisable.

Exercice 1

- 4) On considère l'ordonnancement de T_i , T_j , T_k suivant :
 - $R_i(A)$ $W_k(A)$ $R_j(A)$ $W_i(B)$ $R_j(B)$ $W_k(B)$
 - A) Est-ce cet ordonnancement est sérialisable ?
 - B) Décrire comment le mécanisme d'accès par verrouillage à deux phases sérialise cet ordonnancement ? Déduire l'ordonnancement série équivalent à cette sérialisation?

Exercice 1

- A) Est-ce cet ordonnancement est sérialisable ?
- $Ri(A) \ Wk(A) \ Rj(A) \ Wi(B) \ Rj(B) \ Wk(B)$
- **Graphe de précédence:**
- Conflit sur A: $Ri(A) - Wk(A)$ (Ti précède Tk)
 $Wk(A) - Rj(A)$ (Tk précède Tj)
- Conflit sur B: $Wi(B) - Rj(B)$ (Ti précède Tj)
 $Wi(B) - Wk(B)$ (Ti précède Tk)
- $Rj(B) - Wk(B)$ (Tj précède Tk)
- Il y a un cycle dans ce graphe, cet ordonnancement est alors **non sérialisable**.



Exercice 1

- B) Décrire comment le mécanisme d'accès par verrouillage à deux phases sérialise cet ordonnancement ? Déduire l'ordonnancement série équivalent à cette sérialisation?
- V2P: Xlock verrou exclusif: write
- Slock verrou partagé: read
- Si la transaction demande un slock et que le verrou déjà placé par une autre transaction sur le même granule est un slock alors la demande est satisfaite, sinon le verrou déjà placé est un xlock alors la transaction doit attendre

Exercice 1

- $Ri(A)$ $Wk(A)$ $Rj(A)$ $Wi(B)$ (fin de Ti) $Rj(B)$ (fin de Tj) $Wk(B)$ (fin de Tk)
- Ti : $Ri(A)$ **Slock (A)**, l'obtient
- Tk : $Wk(A)$ **Xlock (A)**, non attente de Ti
- Tj : $Rj(A)$ **Slock (A)**, l'obtient
- Ti : $Wi(B)$ **Xlock (B)**, l'obtient, fin de Ti , $Unlock(A)$, $Unlock(B)$, réveiller Tk
- Tk : $Wk(A)$ **Xlock (A)**, non attente de Tj
- Tj : $Rj(B)$ **Slock (B)**, l'obtient, fin de Tj , $Unlock(A)$, $Unlock(B)$, réveiller Tk
- Tk : $Wk(A)$ **Xlock (A)**, l'obtient
- Tk : $Wk(B)$ **Xlock (B)**, l'obtient, fin de Tk , $Unlock(A)$, $Unlock(B)$
- L'ordonnancement série équivalent à cette sérialisation: Ti Tj Tk

Exercice

Considérons l'ordonnancement O des transactions T_1 , T_2 et T_3

$O: R_3(B) R_1(A) R_2(C) W_1(C) W_2(B) W_2(C) W_3(A)$

1. Vérifier si O est sérialisable en identifiant les conflits et en construisant le graphe de précédence

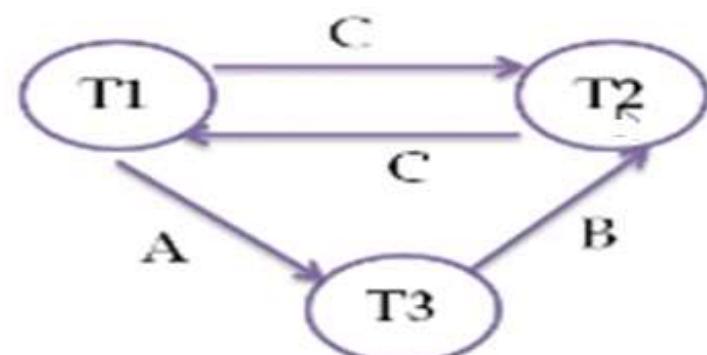
■ Les conflits :

sur B : $R_3(B)$ - $W_2(B)$

sur A : $R_1(A)$ - $W_3(A)$

Sur C : $R_2(C)$ - $W_1(C)$, $W_1(C)$ - $W_2(C)$

■ Graphe de précédence



Le graphe de précédence est cyclique car on a le cycle $\{T_1, T_2, T_1\}$ d'où O n'est pas sérialisable.

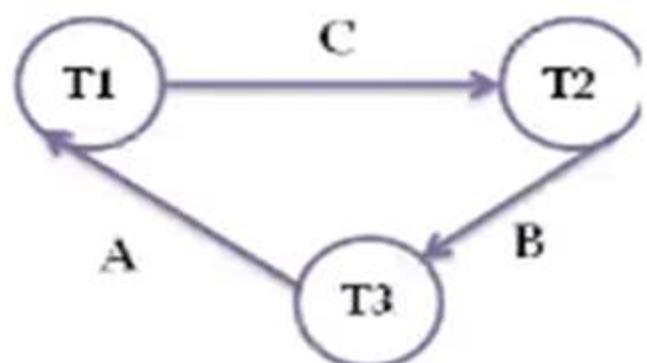
O: R3(B) R1(A) R2(C) W1(C) W2(B) W2(C) W3(A)

2. Appliquer le protocole de verrouillage à deux phases sur O jusqu'à l'exécution de toutes les actions. Quelle est l'exécution finale obtenue ?

Transaction	Action	Demande Verrou	Réponse
T3	R3(B)	SLOCK(B)	OK
T1	R1(A)	SLOCK(A)	OK
T2	R2(C)	SLOCK(C)	OK
T1	W1(C)	XLOCK(C)	Non, T1 attend la fin de T2
T2	W2(B)	XLOCK(B)	Non, T2 attend la fin de T3
T2	W2(C)	XLOCK(C)	T2 est en attente
T3	W3(A)	XLOCK(A)	Non, T3 attend la fin de T1

■ Graphe d'attente

- ✓ De graphe d'attente, on constate qu'on a un cycle T1-T2-T3-T1 d'où on a un inter-blockage.
- ✓ T3 est la victime car elle est la plus récente.
- ✓ Rollback(T3) → annuler T3 et libérer les verrous(unlock(B));
- ✓ Reprendre T2 à partir de point de blockage;
- ✓ T3 sera relancée dès son début à la fin.



O: R3(B) R1(A) R2(C) W1(C) W2(B) W2(C) W3(A)

2. Appliquer le protocole de verrouillage à deux phases sur O jusqu'à l'exécution de toutes les actions. Quelle est l'exécution finale obtenue ?

Transaction	Action	Demande Verrou	Réponse
Rollback(T3): UNLOCK(B) et reprise de T2			
T2	W2(B)	XLOCK(B)	OK
T2	W2(C)	XLOCK(C)	OK, fin de T2 d'où UNLOCK(C), UNLOCK(B), réveiller T1.
Reprise de T1			
T1	W1(C)	XLOCK(C)	OK, fin de T1, UNLOCK(A), UNLOCK(C).
Relancer T3 dès son début :			
T3	R3(B)	SLOCK(B)	OK
T3	W3(A)	XLOCK(A)	OK, fin de T3, UNLOCK(A), UNLOCK(B).

- L'exécution finale obtenue est T2, T1, T3