

Contrôle de concurrence Quelques exercices CORRIGE

Exercice 1.

Quelles sont parmi les exécutions suivantes celles qui produisent des anomalies ?

ai signifie que la transaction i est défaite

ci signifie que la transaction i est validée

1. $r1(x), w1(x), r2(x), w2(y), a1, c2$

anomalie de lecture erronée

la transaction 2 lit une valeur de x mise à jour par la transaction 1 mais qui ne sera finalement pas validée (a1)

2. $r1(x), w1(x), r2(y), w2(y), a1, c2$

pas de problème

les transactions 1 et 2 n'agissent pas sur le même objet

3. $r1(x), r2(x), r2(y), w2(y), r1(z), a1, c2$

pas de problème

x, objet concurrent, n'est que lu par les deux transactions

4. $r1(x), r2(x), w2(x), w1(x), c1, c2$

anomalie de perte de mise à jour

la mise à jour faite par la transaction 2 est « écrasée » par la mise à jour faite par 1 sur une valeur de x antérieure à l'action de 2

5. $r1(x), r2(x), w2(x), r1(y), c1, c2$

pas de problème

6. $r1(x), w1(x), r2(x), w2(x), c1, c2$

pas de problème (exécution en série)

Exercice 2.

Quelles sont, ci-dessous, les exécutions vue-sérialisables ?

1. $r1(x), r2(y), w1(y), r2(x), w2(x)$: NON

2. $r1(x), r2(y), w1(x), w1(y), r2(x), w2(x)$: NON

3. $r1(x), r1(y), r2(y), w2(z), w1(z), w3(z), w3(x)$: OUI, car vue-équivalente à 1-2-3

4. $r1(y), r1(y), w2(z), w1(z), w3(z), w3(x), w1(x)$: NON

Exercice 3.

Classifier les exécutions suivantes (non vue-sérialisables, vue-sérialisables, conflit-sérialisables). En cas d'équivalence, donnez toutes les exécutions séries équivalentes.

1. $r1(x), w1(x), r2(z), r1(y), w1(y), r2(x), w2(x), w2(z)$

vue-sérialisable car vue-équivalente avec 1-2

conflit-sérialisable car conflit-équivalente avec 1-2

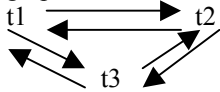
graphe de conflit-sérialisabilité : $t1 \rightarrow t2$

2. $r1(x), w1(x), w3(x), r2(y), r3(y), w3(y), w1(y), r2(x)$

non vue-sérialisable

non conflit-sérialisable

graphe de conflit-sérialisabilité :

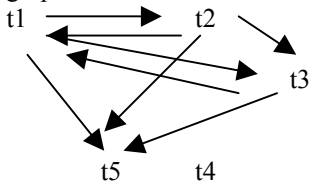


3. $r1(x), r2(x), w2(x), r3(x), r4(z), w1(x), w3(y), w3(x), w1(y), w5(x), w1(z), w5(y), r5(z)$

non vue-sérialisable

non conflit-sérialisable

graphe de conflit-sérialisabilité :

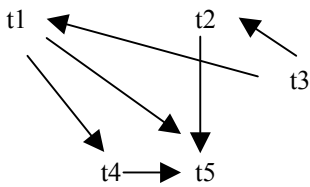


4. $r1(x), r3(y), w1(y), w4(x), w1(t), w5(x), r2(z), r3(z), w2(z), w5(z), r4(t), r5(t)$

vue-sérialisable car vue-équivalente avec 3-1-2-4-5, 3-1-4-2-5, 3-2-1-4-5

conflit-sérialisable avec 3-2-1-4-5

graphe de conflit-sérialisabilité



5. $r_1(x), r_2(x), w_2(x), r_3(x), r_4(z), w_1(x), r_3(y), r_3(x), w_1(y), w_5(x), w_1(z), r_5(y), r_5(z)$

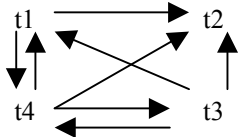
la théorie ne s'applique pas car la transaction 3 lit deux fois x

6. $r_1(x), r_1(t), r_3(z), r_4(z), w_2(z), r_4(x), r_3(x), w_4(x), w_4(y), w_3(y), w_1(y), w_2(t)$

non vue-sérialisable

non conflit-sérialisable

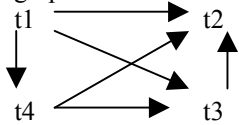
graphe de conflit-sérialisabilité



7. $r_1(x), r_4(x), w_4(x), r_1(y), r_4(z), w_4(z), w_3(y), w_3(z), w_1(t), w_2(z), w_2(t)$

conflit-équivalente et vue-équivalente avec 1-4-3-2

graphe de conflit-sérialisabilité



Exercice 4.

Pour chacune des exécutions ci-dessus (exercice 3), décrire les transactions qui seraient placées en attente dans le cas d'un verrouillage deux phases. (On ne considèrera pas la suite des actions d'une transaction mise en attente)

Exécution 1

$r_lock1(x)$

$r_1(x)$

$w_lock1(x)$

$w_1(x)$

$r_lock2(z)$

$r_2(z)$

$r_lock1(y)$

$r_1(y)$

$w_lock1(y)$

$w_1(y)$

$r_lock2(x)$

mise en attente de 2

$unlock1(x)$

$unlock1(y)$

2 peut reprendre

Exécution 2

$r_lock1(x)$

$r_1(x)$

$w_lock1(x)$

$w_1(x)$

$r_lock3(x)$

mise en attente de 3

$r_lock2(y)$

$r_2(y)$

$w_lock1(y)$

1 n'attend pas 2

1 est défaite et reprend avec le numéro 4

Exécution 3

$r_lock1(x)$

$r_1(x)$

$r_lock2(x)$

$r_2(x)$

$w_lock2(x)$

mise en attente de 2
r_lock3(x)
r3(x)
r_lock4(z)
r4(z)
w_lock1(x)
1 n'attend pas 2
1 est défaite et reprend avec le numéro 5
w_lock3(y)
w3(y)
w_lock3(x)
....

Exercice 5.

Si les exécutions de l'exercice 3 étaient présentées à un système d'estampillage, quelles transactions seraient défaits ?

Exécution 1

r1(x) OK RTM(x)=1
w1(x) OK WTM(x)=1
r2(z) OK RTM(z)=2
r1(y) OK RTM(y)=1
w1(y) OK RTM(y)=1
r2(x) OK RTM(x)=2
w2(x) OK WTM(x)=2
w2(z) OK WTM(z)=2
aucune transaction défaite

Exécution 2

r1(x) OK RTM(x)=1
w1(x) OK WTM(x)=1
w3(x) OK WTM(x)=3
r2(y) OK RTM(y)=3
r3(y) OK RTM(y)=3
w3(y) OK WTM(y)=3
w1(y) non => transaction 1 défaite
r2(x) non => transaction 2 défaite

Exécution 3

r1(x) OK RTM(x)=1
r2(x) OK RTM(x)=2
w2(x) OK WTM(x)=2
r3(x) OK RTM(x)=3
r4(z) OK RTM(z)=4
w1(x) non => transaction 1 défaite
w3(y) OK WTM(y)=3
w3(x) OK WTM(x)=3
w5(x) OK WTM(x)=5
w5(y) OK WTM(y)=5
r5(z) OK RTM(z)=5

Exécution 4

r1(x) OK RTM(x)=1
r3(y) OK RTM(y)=3
w1(y) non => transaction 1 défaite
w4(x) OK WTM(x)=4
w5(x) OK WTM(x)=5
r2(z) OK RTM(z)=2
r3(z) OK RTM(z)=3
w2(z) non => transaction 2 défaite
w5(z) OK WTM(z)=5
r4(t) OK RTM(t)=4
r5(t) OK RTM(t)=5

Exécution 5

r1(x) OK RTM(x)=1
r2(x) OK RTM(x)=2
w2(x) OK WTM(x)=2
r3(x) OK RTM(x)=3
r4(z) OK RTM(z)=4

w1(x) non => transaction 1 défaite
r3(y) OK RTM(y)=3
r3(x) OK RTM(x)=3
w5(x) OK WTM(x)=5
r5(y) OK RTM(y)=5
r5(z) OK RTM(z)=5

Exécution 6

r1(x) OK RTM(x)=1
r1(t) OK RTM(t)=1
r3(z) OK RTM(z)=3
r4(z) OK RTM(z)=4
w2(z) non => transaction 2 défaite
r4(x) OK RTM(x)=4
r3(x) OK RTM(x)=4
w4(x) OK WTM(x)=4
w4(y) OK WTM(y)=4
w3(y) non => transaction 3 défaite
w1(y) non => transaction 1 défaite

Exécution 7

r1(x) OK RTM(x)=1
r4(x) OK RTM(x)=4
w4(x) OK WTM(x)=4
r1(y) OK RTM(y)=1
r4(z) OK RTM(z)=4
w4(z) OK WTM(z)=4
w3(y) OK WTM(y)=3
w3(z) non => transaction 3 défaite
w1(t) OK WTM(t)=1
w2(z) non => transaction 2 défaite