TD N° 4 - Correction
La Concurrence

Exercice 1 : Recouvrabilité

Pour chacun des historiques ci-dessous, indiquez s'il est recouvrable, s'il évite les annulations en cascade et s'il est un historique strict.

H_1 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_2(x)$ $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 C_2 $W_1(z)$ C_1

H_2 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_2(x)$ $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 $W_1(z)$ C_1 C_2

H_3 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_2(x)$ $W_1(z)$ C_1 $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 C_2

H_4 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_1(z)$ C_1 $W_2(x)$ $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 C_2

Solution :

H_1 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_2(x)$ $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 C_2 $W_1(z)$ C_1

H_1 n'est pas recouvrable car T_2 lit y à partir de T_1 et T_2 valide avant T_1.

Donc H_1 n'évite pas les annulations en cascade et n'est pas strict.

H_2 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_2(x)$ $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 $W_1(z)$ C_1 C_2

H_2 est recouvrable car T_2 lit y à partir de T_1 et T_2 valide après T_1.

H_2 n'évite pas les annulations en cascade car T_2 lit y à partir de T_1 et T_1 ne valide pas avant la lecture faite par T_2.

Donc H_2 n'est pas strict

H_3 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_2(x)$ $W_1(z)$ C_1 $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 C_2

H_3 est recouvrable car T_2 lit y à partir de T_1 et T_2 valide après T_1.

H_3 évite les annulations en cascade car T_2 lit y à partir de T_1 et T_1 valide avant la lecture faite par T_2.

H_3 n'est pas une exécution stricte car T_1 modifie x puis T_2 modifie x avant que T_1 valide.

H_4 : $W_1(x)$ $W_1(y)$ $R_2(u)$ $W_1(z)$ C_1 $W_2(x)$ $R_2(y)$ $W_2(y)$ $W_3(u)$ C_3 C_2

H_4 est recouvrable car T_2 lit y à partir de T_1 et T_2 valide après T_1.

H_4 évite les annulations en cascade car T_2 lit y à partir de T_1 et T_1 valide avant la lecture faite par T_2.

H_4 est une exécution stricte car les lectures et les écritures faites par T_2 sur des données modifiées par T_1 ont lieu après la validation de T_1.

Exercice 2 : Graphe de précédence

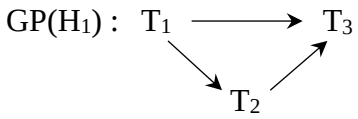
Pour chacun des historiques ci-dessous, tracer le graphe de précédence. En déduire lorsque s'il y a lieu, les historiques sériels équivalents.

H_1 : $R_1(x)$ $W_1(y)$ $R_2(y)$ $W_2(z)$ $R_3(z)$ $W_3(x)$ C_1 C_2 C_3

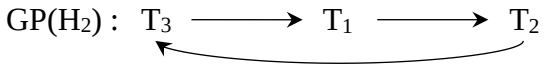
H_2 : $W_3(x)$ $R_1(x)$ $W_1(y)$ $R_2(y)$ $W_2(z)$ $R_3(z)$ C_1 C_2 C_3

Solution :

H_1 est conflit-sérialisable car conflit-équivalent à $T_1T_2T_3$



H_2 n'est pas conflit-sérialisable car $GP(H_2)$ possède un cycle.



Exercice 3 : Verrouillage à 2 phases

Pour chacun des historiques de l'exercice précédent, poser les verrous en retardant le plus possible la suppression des verrous d'écriture et indiquer si l'historique obtenu est conforme au protocole 2PL. Dans l'affirmative, indiquer si l'historique est conforme à S2PL.

1) Pose de verrous sur H_1 :

H_1' : $SL_1(x)\ R_1(x)\ XL_1(y)\ W_1(y)\ U_1(y)\ SL_2(y)\ R_2(y)\ XL_2(z)\ W_2(z)\ U_2(z)\ SL_3(z)\ R_3(z)\ U_1(x)\ XL_3(x)\ W_3(x)\ C_1\ U_2(y)\ C_2\ U_3(z)\ C_3$

T_1' : $SL_1(x)\ R_1(x)\ XL_1(y)\ W_1(y)\ U_1(y)\ U_1(x)\ C_1$
T_2' : $SL_2(y)\ R_2(y)\ XL_2(z)\ W_2(z)\ U_2(z)\ U_2(y)\ C_2$
T_3' : $SL_3(z)\ R_3(z)\ U_3(z)\ C_3$

T_1', T_2' et T_3' sont des transactions à 2 phases. Donc H_1' est conforme à 2PL.
H_1' n'est pas conforme à S2PL car T_1' dans H_1' relâche son verrou d'écriture $U_1(y)$ avant d'atteindre C_1 (idem pour T_2' et $U_2(z)$). Donc H_1' (et H_1) n'est pas strict.
De plus, il n'évite pas les annulations en cascade car T_3 lit à partir de T_2 et $R_3(z) < C_2$, mais il est recouvrable.

3) Pose de verrous sur H_2 :

H_2' : $XL_3(x)\ W_3(x)\ U_3(x)\ SL_1(x)\ R_1(x)\ XL_1(y)\ W_1(y)\ U_1(y)\ SL_2(y)\ R_2(y)\ XL_2(z)\ W_2(z)\ U_2(z)\ SL_3(z)\ R_3(z)\ C_1\ U_1(x)\ C_2\ U_2(y)\ U_3(z)\ C_3$

T_1' : $SL_1(x)\ R_1(x)\ XL_1(y)\ W_1(y)\ U_1(y)\ U_1(x)\ C_1$
T_2' : $SL_2(y)\ R_2(y)\ XL_2(z)\ W_2(z)\ U_2(z)\ U_2(y)\ C_2$
T_3' : $XL_3(x)\ W_3(x)\ U_3(x)\ SL_3(z)\ R_3(z)\ U_3(z)\ C_3$

T_1' et T_2' sont des transactions à 2 phases, mais T_3' n'est pas à 2 phases (voir fond gris).
Donc H_2'n'est pas conforme à 2PL.

Exercice 4 : Protocole avec estampille

Le tableau ci-dessous illustre l'exécution concurrente (avec estampillage) de trois transactions qui accèdent aux 3 données x, y, et z. Compléter ce tableau :

- en mettant à jour les estampilles tsR (timestamp Reader) et tsW (timestamp Writer) associées à chaque donnée,
- en insérant l'instruction **abort** si les actions R_1, R_2, R_3, W_1, W_2, W_3 ne peuvent s'exécuter.

Solution :

T_1 ts=200	T_2 ts=150	T_3 ts=175	x tsR=0, tsW=0	y tsR=0, tsW=0	z tsR=0, tsW=0
$R_1(y)$				tsR = 200	
	$R_2(x)$		tsR = 150		
		$R_3(z)$			tsR = 175
$W_1(y)$				tsW = 200	
$W_1(x)$			tsW = 200		
	$W_2(z)$ **Abort**				tsW = ?
		$W_3(x)$ **Abort**	tsW = ?		

Exercice 5 : Sérialisabilité et Multiversion

Pour chacun des historiques ci-dessous, indiquer sa sérialisabilité :

- est-il vue-sérialisable,
- est-il conflit-sérialisable,
- ou peut-il être traduit par un historique multi-version qui soit 1V-sérialisable.

Pour chaque historique qui n'est pas sérialisable (ni vue-sérialisable, ni conflit-sérialisable, ni 1V-sérialisable), indiquer quelles sortes d'anomalies il présente.

H_1 : $W_0(x)\ C_0\ R_1(x)\ C_1\ R_2(x)\ W_2(x)\ W_2(z)\ C_2$

H_2 : $W_0(x)\ C_0\ R_2(x)\ R_1(x)\ C_1\ W_2(x)\ W_2(z)\ C_2$

H_3 : $W_0(x)\ C_0\ R_1(x)\ R_2(x)\ W_2(x)\ C_2\ W_1(x)\ C_1$

H_4 : $W_0(x)\ C_0\ R_{1.1}(x)\ R_2(x)\ W_2(x)\ R_{1.2}(x)\ C_1\ C_2$

Remarque : H_4 sort du modèle simple présenté en cours. Modifier la définition des lectures « à partir de » afin de décrire H_4.

Solution :

H_1 est sériel ($H_1=T_0T_1T_2$) donc H_1 est trivialement vue-sérialisable et conflit-sérialisable.

On peut traduire H_1 par 1 seul historique MV :

H_{MV1} : $W_0(x_0)\ C_0\ R_1(x_0)\ C_1\ R_2(x_0)\ W_2(x_2)\ W_2(z_2)\ C_2$

H_{MV1} est équivalent à H_1. De plus H_{MV1} est 1V-sériel car :

- H_{MV1} est sériel ($T_0T_1T_2$)
- T_1 lit la dernière version de x (écrite par T_0)
- T_2 lit aussi la dernière version de x (écrite par T_0).

Donc H_{MV1} est trivialement 1V-sérialisable.

H_2 est conflit-sérialisable car H_2 est conflit-équivalent à H_1. En effet :

$OC(H_2) = OC(H_1) = \{ W_0(x) < R_1(x) ; W_0(x) < R_2(x) ; W_0(x) < W_2(x) ; R_1(x) < W_2(x) \}$

H_2 est vue-sérialisable car H_2 est conflit-sérialisable.

On peut traduire H_2 par 1 seul historique MV :

H_{MV2} : $W_0(x_0)\ C_0\ R_2(x_0)\ R_1(x_0)\ C_1\ W_2(x_2)\ W_2(z_2)\ C_2$

H_{MV2} est équivalent à H_{MV1} (ils ont exactement les mêmes opérations) et H_{MV1} est 1V-sériel. Donc H_{MV2} est 1V-sérialisable.

H_3 n'est pas vue-sérialisable car

1) Dans $T_0T_1T_2$, T_2 lit x à partir de T_1, ce qui n'est pas vrai dans H_3
2) Dans $T_0T_2T_1$, T_1 lit x à partir de T_2, ce qui n'est pas vrai dans H_3

Donc $RF(H_3) \neq RF(T_0T_1T_2)$ et $RF(H_3) \neq RF(T_0T_2T_1)$

H_3 n'est pas conflit-sérialisable car H_3 n'est pas vue-sérialisable.

On peut traduire H_3 par 1 seul historique MV :

H_{MV3} : $W_0(x_0)\ C_0\ R_1(x_0)\ R_2(x_0)\ W_2(x_2)\ C_2\ W_1(x_1)\ C_1$

H_{MV3} n'est pas 1V-sérialisable car :

H_{T0T1T2} : $W_0(x_0)\ C_0\ R_1(x_0)\ W_1(x_1)\ C_1\ R_2(x_0)\ W_2(x_2)\ C_2$ n'est pas 1V-sériel

H_{T0T2T1} : $W_0(x_0)\ C_0\ R_2(x_0)\ W_2(x_2)\ C_2\ R_1(x_0)\ W_1(x_1)\ C_1$ n'est pas 1V-sériel

H_3 présente une anomalie de mise à jour perdue

H_4 n'est pas vue-sérialisable car

Dans $T_0T_1T_2$, T_1 lit **la deuxième fois** x à partir de T_0 (i.e. $W_2(x) < R_{1.2}(x)$), ce qui n'est pas vrai dans H_4.
Dans $T_0T_2T_1$, T_1 lit **la première fois** x à partir de T_2 (i.e. $W_2(x) < R_{1.1}(x)$), ce qui n'est pas vrai dans H_4.
Donc $RF(H_4) \neq RF(T_0T_1T_2)$ et $RF(H_4) \neq RF(T_0T_2T_1)$

H_4 n'est pas conflit-sérialisable car H_4 n'est pas vue-sérialisable.

On peut traduire H_4 par 2 historiques MV :

$H_{MV4.1}$: $W_0(x_0)\ C_0\ R_{1.1}(x_0)\ R_2(x_0)\ W_2(x_2)\ R_{1.2}(x_2)\ C_1\ C_2$

$H_{MV4.2}$: $W_0(x_0)\ C_0\ R_{1.1}(x_0)\ R_2(x_0)\ W_2(x_2)\ R_{1.2}(x_0)\ C_1\ C_2$

$H_{MV4.1}$ n'est pas 1V-sérialisable car :

H_{T0T1T2} : $W_0(x_0)\ C_0\ R_{1.1}(x_0)\ R_{1.2}(x_2)\ C_1\ R_2(x_0)\ W_2(x_2)\ C_2$ n'est pas un historique MV

H_{T0T2T1} : $W_0(x_0)\ C_0\ R_2(x_0)\ W_2(x_2)\ C_2\ R_{1.1}(x_0)\ R_{1.2}(x_2)\ C_1$ n'est pas 1V-sériel

$H_{MV4.2}$ est 1V-sérialisable car :

H_{T0T1T2} : $W_0(x_0)\ C_0\ R_{1.1}(x_0)\ R_{1.2}(x_0)\ C_1\ R_2(x_0)\ W_2(x_2)\ C_2$ est 1V-sériel

$H_{MV4.2}$ est équivalent à H_{T0T1T2}

Donc H_4 est sérialisable car il peut être traduit en un historique MV qui 1V-sérialisable.