

11. Serializzazione

Esercizi

Esercizio

- Indicare se i seguenti schedule sono VSR:

1. $r1(x), r2(y), w1(y), r2(x), w2(x)$
2. $r1(x), r2(y), w1(x), w1(y), r2(x), w2(x)$
3. $r1(x), r1(y), r2(y), w2(z), w1(z), w3(z), w3(x)$
4. $r1(x), r4(x), w4(x), r1(y), r4(z), w4(z), w3(y),$
 $w3(z), w1(t), w2(z), w2(t)$

Esercizio

- Nei primi due casi abbiamo solo due possibili schedule seriali:
 - S1: $r1(x), w1(y), r2(y), r2(x), w2(x)$
 - $LEGGE-DA(S1) = \{(w1(y), r2(y))\}$
 - $FINALE = \{w1(y), w2(x)\}$
 - S2: $r2(y), r2(x), w2(x), r1(x), w1(y)$
 - $LEGGE-DA(S2) = \{(w2(x), r1(x))\}$
 - $FINALE = \{w1(y), w2(x)\}$

Esercizio

- $\text{LEGGE-DA}(S1) = \{(w1(y), r2(y))\}$
- $\text{LEGGE-DA}(S2) = \{(w2(x), r1(x))\}$
- Lo schedule $r1(x), r2(y), w1(y), r2(x), w2(x)$:
 - $\text{LEGGE-DA} = \{\}$
- quindi non è VSR perché non è view-equivalente né a $S1$ né a $S2$

Esercizio

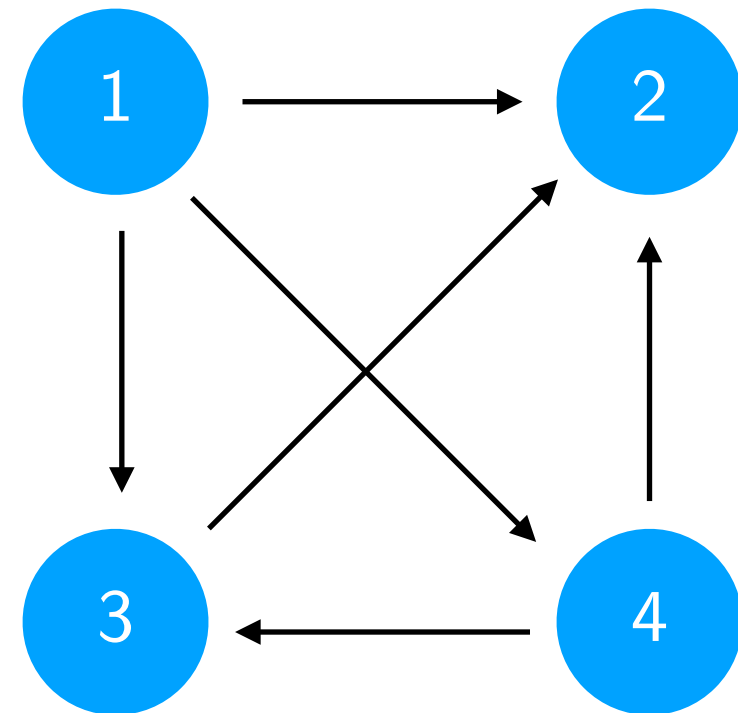
- $\text{LEGGE-DA}(S1) = \{(w1(y), r2(y))\}$
- $\text{LEGGE-DA}(S2) = \{(w2(x), r1(x))\}$
- Lo schedule $r1(x), r2(y), w1(x), w1(y), r2(x), w2(x)$:
 - $\text{LEGGE-DA} = \{(w1(x), r2(x))\}$
- quindi non è VSR perché non è view-equivalente né a $S1$ né a $S2$

Esercizio

- $r1(x), r1(y), r2(y), w2(z), w1(z), w3(z), w3(x)$
 - $LEGG\text{-}DA = \{\}$
 - $FINALE = \{w3(x), w3(z)\}$
- Per essere VSR dobbiamo trovare uno schedule seriale a cui è view-equivalente
 - Deve avere uguale FINALE, quindi T3 deve essere l'ultima transazione
 - Schedule seriali:
 - $r2(y), w2(z), r1(x), r1(y), w1(z), w3(z), w3(x)$
 - $LEGG\text{-}DA = \{\}$
 - $r1(x), r1(y), w1(z), r2(y), w2(z), w3(z), w3(x)$
 - $LEGG\text{-}DA = \{\}$
- Quindi è VSR

Esercizio

- $r1(x)$, $r4(x)$, $w4(x)$, $r1(y)$, $r4(z)$, $w4(z)$, $w3(y)$, $w3(z)$, $w1(t)$, $w2(z)$, $w2(t)$
- Questa è difficile, ci sono 4 transazioni su 4 oggetti
- Proviamo a verificare se è CSR
 - x : $r1$, $r4$, $w4$
 - y : $r1$, $w3$
 - z : $r4$, $w4$, $w3$, $w2$
 - t : $w1$, $w2$
- Il grafo dei conflitti è aciclico
- Lo schedule è CSR, quindi anche VSR



Esercizio

- Indicare se i seguenti schedule possono produrre anomalie:

1. $r1(x), w1(x), r2(x), w2(y), a1, c2$
2. $r1(x), w1(x), r2(y), w2(y), a1, c2$
3. $r1(x), r2(x), r2(y), w2(y), r1(z), a1, c2$
4. $r1(x), r2(x), w2(x), w1(x), c1, c2$
5. $r1(x), r2(x), w2(x), r1(y), c1, c2$
6. $r1(x), w1(x), r2(x), w2(x), c1, c2$

Esercizio

- $r1(x)$, $w1(x)$, $r2(x)$, $w2(y)$, $a1$, $c2$
- L'operazione $r2(x)$ legge il valore scritto da $w1(x)$, ma la transazione T1 termina con un abort.
- Questo è un caso di lettura sporca (Dirty Read) e anche la transazione T2 deve essere abortita.

Esercizio

- $r1(x)$, $w1(x)$, $r2(y)$, $w2(y)$, $a1$, $c2$
- Questo schedule non produce anomalie, perché le due transazioni fanno riferimento a oggetti differenti.

Esercizio

- $r1(x)$, $r2(x)$, $r2(y)$, $w2(y)$, $r1(z)$, $a1$, $c2$
- Questo schedule non produce anomalie, perché la transazione T1 che termina in abort non effettua operazioni di scrittura.

Esercizio

- $r1(x)$, $r2(x)$, $w2(x)$, $w1(x)$, $c1$, $c2$
- Questo schedule ha una perdita di aggiornamento (lost update), in quanto gli effetti della transazione T2 vengono persi.

Esercizio

- $r1(x)$, $r2(x)$, $w2(x)$, $r1(y)$, $c1$, $c2$
- Questo schedule non produce anomalie.

Esercizio

- $r1(x)$, $w1(x)$, $r2(x)$, $w2(x)$, $c1$, $c2$
- Questo schedule non produce anomalie.

Esercizio

- Se il seguente schedule si presentasse a uno scheduler che usa il locking a due fasi, quali transazioni verrebbero messe in attesa?
 - Una volta posta in attesa una transazione, le sue successive azioni non vanno più considerate.
- $r1(x)$, $r3(y)$, $w1(y)$, $w4(x)$, $w1(t)$, $w5(x)$, $r2(z)$, $r3(z)$, $w2(z)$, $w5(z)$, $r4(t)$, $r5(t)$

Esercizio

- $r1(x)$, $r3(y)$, $w1(y)$, $w4(x)$, $w1(t)$, $w5(x)$, $r2(z)$, $r3(z)$, $w2(z)$, $w5(z)$, $r4(t)$, $r5(t)$
- Le transazioni 1, 4, 5 e 2 sono in attesa.
 - La transazione 1 deve aspettare per y (allocato da 3)
 - Le transazioni 4 e 5 devono aspettare per x (allocato da 1)
 - La transazione 2 deve aspettare per z (allocato da 3).
- Consideriamo che tutte le transazioni terminino con successo, qual è un possibile ordine dei commit?
 - $r1(x)$, $r3(y)$, $w1(y)$, $w4(x)$, $w1(t)$, $w5(x)$, $r2(z)$, $r3(z)$, $c3$, $w2(z)$, $c2$, $c1$, $w5(z)$, $r4(t)$, $c4$, $r5(t)$, $c5$

Esercizio

- Se il seguente schedule si presentasse a uno scheduler che usa il locking a due fasi, quali transazioni verrebbero messe in attesa?
- Una volta posta in attesa una transazione, le sue successive azioni non vanno più considerate.
- $r1(x)$, $r2(x)$, $w2(x)$, $r3(x)$, $r4(z)$, $w1(x)$, $r3(y)$, $r3(x)$, $w1(y)$, $w5(x)$, $w1(z)$, $r5(y)$, $r5(z)$

Esercizio

- $r1(x)$, $r2(x)$, $w2(x)$, $r3(x)$, $r4(z)$, $w1(x)$, $r3(y)$, $r3(x)$, $w1(y)$, $w5(x)$, $w1(z)$, $r5(y)$, $r5(z)$
- Le transazioni 2, 1 e 5 sono in attesa. Devono aspettare per x (allocata da 1, 2 e 3), 3 termina ma le altre no

Esercizio

- Se il seguente schedule si presentasse a uno scheduler che usa il locking a due fasi, quali transazioni verrebbero messe in attesa?
- Una volta posta in attesa una transazione, le sue successive azioni non vanno più considerate.
- $r1(x)$, $r1(t)$, $r3(z)$, $r4(z)$, $w2(z)$, $r4(x)$, $r3(x)$, $w4(x)$, $w4(y)$, $w3(y)$, $c3$, $w1(y)$, $c1$, $w2(t)$, $c2$, $c4$

Esercizio

- $r1(x)$, $r1(t)$, $r3(z)$, $r4(z)$, $w2(z)$, $r4(x)$, $r3(x)$, $w4(x)$, $w4(y)$, $w3(y)$, $c3$, $w1(y)$, $c1$, $w2(t)$, $c2$, $c4$
- Le transazioni 2 e 4 vengono messe in attesa. 2 per z (allocata da 3) e 4 per x (allocata da 1 e 3)

Esercizio

- Se il seguente schedule si presentasse a uno scheduler che usa il locking a due fasi, quali transazioni verrebbero messe in attesa?
- Una volta posta in attesa una transazione, le sue successive azioni non vanno più considerate.
- $r1(x)$, $r4(x)$, $w4(x)$, $r1(y)$, $r4(z)$, $w4(z)$, $w3(y)$, $w3(z)$, $w1(t)$, $c1$, $w2(z)$, $w2(t)$, $c2$, $c4$, $c3$

Esercizio

- $r1(x)$, $r4(x)$, $w4(x)$, $r1(y)$, $r4(z)$, $w4(z)$, $w3(y)$, $w3(z)$, $w1(t)$, $c1$, $w2(z)$, $w2(t)$, $c2$, $c4$, $c3$
- Le transazioni 3 e 4 sono in attesa. Devono aspettare per x e y , allocate da 1.

Esercizio

- Se il seguente schedule si presentasse a uno scheduler basato su timestamp, quali transazioni verrebbero abortite?
 - RTM e WTM valgono 0 per ogni risorsa all'inizio.
 - I TS corrispondono al numero della transazione.
- $r1(x), r3(y), w1(y), w4(x), w1(t), w5(x), r2(z), r3(z), w2(z), w5(z), r4(t), r5(t)$
-

Esercizio

OPERAZIONE	RISPOSTA	NUOVO VALORE
read(x, 1)	OK	RTM(x) = 1
read(y, 3)	OK	RTM(y) = 3
write(y, 1)	T1 abortita	
write(x, 4)	OK	WTM(x) = 4
write(x, 5)	OK	WTM(x) = 5
read(z, 2)	OK	RTM(z) = 2
read(z, 3)	OK	RTM(z) = 3
write(z, 2)	T2 abortita	
write(z, 5)	OK	WTM(z) = 5
read(t, 4)	OK	RTM(t) = 4
read(t, 5)	OK	RTM(x) = 5