

Esercizi su Schedule

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. $r1(x), r1(y), r2(y), w2(z), w1(z), w3(z), w3(x)$

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

- Questo schedule non è VSR, perché i due schedule seriali:

- S1: $r1(x), w1(y), r2(y), r2(x), w2(x)$



- S2: $r2(y), r2(x), w2(x), r1(x), w1(y)$

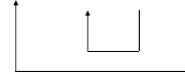


non sono view-equivalenti con lo schedule dato.
Hanno entrambi una differente relazione LEGGE - DA

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

- Questo schedule non è VSR perché gli schedule

- S1: $r1(x), w1(x), w1(y), r2(y), r2(x), w2(x)$



- S2: $r2(y), r2(x), w2(x), r1(x), w1(x), w1(y)$



hanno entrambi una differente relazione LEGGE - DA

3. $r1(x), r1(y), r2(y), w2(z), w1(z), w3(z), w3(x)$

- Questo schedule è VSR e view-equivalente allo schedule seriale

- S: $r2(y), w2(z), r1(x), r1(y), w1(z), w3(z), w3(x)$

in quanto sono caratterizzati dalle stesse scritture finali e dalle stesse relazioni LEGGI - DA.

Indicare se i seguenti schedule possono produrre anomalie.
 ci e ai indicano l'esito (commit o abort) della transazione.

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

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

- L'operazione r2(x) legge il valore scritto da w1(x), ma la transazione 1 termina con un abort.
- Questo è un caso di lettura sporca

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

- Questo schedule non produce anomalie, perché le due transazioni fanno riferimento a oggetti differenti.

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

- Questo schedule non produce anomalie, perché la transazione 1 che termina in abort non effettua operazioni di scrittura.

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

- Questo schedule ha una perdita di aggiornamento, in quanto gli effetti della transazione 2 vengono persi.

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

- Questo schedule non produce anomalie.

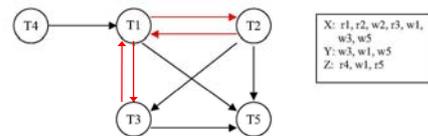
6) $r1(x), w1(x), r2(x), w2(x), c1, c2$

- Questo schedule non produce anomalie

Classificare lo schedule

- $r1(x), r2(x), w2(x), r3(x), r4(z), w1(x), w3(y), w3(x), w1(y), w5(x), w1(z), w5(y), r5(z)$
- Per classificare questo schedule si deve realizzare un grafo dei conflitti:
 - un nodo per ogni transazione t_i
 - un arco (orientato) da t_i a t_j se c'è almeno un conflitto fra un'azione a_i e un'azione a_j tale che a_i precede a_j
 - un'azione a_i è in *conflitto* con a_j ($i \neq j$), se operano sullo stesso oggetto e almeno una di esse è una scrittura.

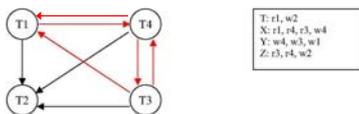
Consideriamo le operazioni relative a ogni risorsa separatamente:



- Questo schedule non è CSR perché il suo grafo dei conflitti è ciclico.

Classificare lo schedule

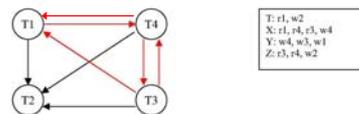
- $r1(x), r1(t), r3(z), r4(z), w2(z), r4(x), r3(x), w4(x), w4(y), w3(y), w1(y), w2(t)$



- Questo schedule non è CSR perché il suo grafo dei conflitti è ciclico.

Classificare lo schedule

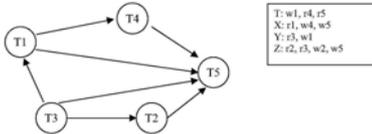
- $r1(x), r1(t), r3(z), r4(z), w2(z), r4(x), r3(x), w4(x), w4(y), w3(y), w1(y), w2(t)$



- Questo schedule non è CSR perché il suo grafo dei conflitti è ciclico.

Classificare lo schedule

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



- Questo schedule è CSR.

Classificare lo schedule

- $r1(x), r3(y), w1(y), w4(x), w1(t), w5(x), r2(z), r3(z), w2(z), w5(z), r4(t), r5(t)$
- Questo schedule è anche VSR. Gli schedule equivalenti sono:
 - S1: $r3(y), r3(z), r1(x), w1(y), w1(t), r2(z), w2(z), w4(x), r4(t), w5(x), w5(z), r5(t)$
 - S2: $r3(y), r3(z), r2(z), w2(z), r1(x), w1(y), w1(t), w4(x), r4(t), w5(x), w5(z), r5(t)$