

Esercizi sui test VSR e CSR

Classificare i seguenti schedule (come: NonSR, VSR, CSR); nel caso uno schedule sia VSR oppure CSR, indicare tutti gli schedule seriali e esso equivalenti.

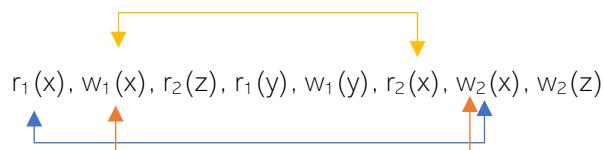
1. $r_1(x), w_1(x), r_2(z), r_1(y), w_1(y), r_2(x), w_2(x), w_2(z)$
2. $r_1(x), w_1(x), w_3(x), r_2(y), r_3(y), w_3(y), w_1(y), r_2(x)$
3. $r_1(x), r_2(x), w_2(x), r_3(x), r_4(z), w_1(x), w_3(y), w_3(x), w_1(y), w_5(x), w_1(z), w_5(y), r_5(z)$
4. $r_1(x), r_3(y), w_1(y), w_4(x), w_1(t), w_5(x), r_2(z), r_3(z), w_2(z), w_5(z), r_4(t), r_5(t)$

SOLUZIONI

$$S_1 = r_1(x), w_1(x), r_2(z), r_1(y), w_1(y), r_2(x), w_2(x), w_2(z)$$

Scelgo di verificare prima CSR.

Calcolo l'insieme dei conflitti di S_1 .



$$\text{Conflitti}(S_1) = \{(r_1(x), w_2(x)), (w_1(x), r_2(x)), (w_1(x), w_2(x))\}$$

Genero il grafo dei conflitti di S_1 e verifico che sia ACICLICO.



Il grafo è ACICLICO e quindi:

S_1 è CSR
(S_1 è conflict-serializzabile)

Poiché CSR \Rightarrow VSR, allora è anche vero che:

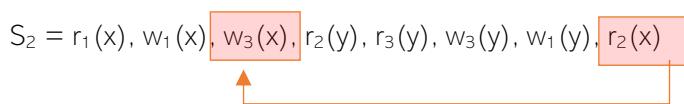
S_1 è VSR

Schedule seriali equivalenti: permutazione (T_1, T_2):

$$\text{SS}_1 = r_1(x), w_1(x), r_1(y), w_1(y), r_2(z), r_2(x), w_2(x), w_2(z)$$

$\underbrace{\qquad\qquad\qquad}_{T_1} \quad \underbrace{\qquad\qquad\qquad}_{T_2}$

SOLUZIONI



Scelgo di verificare prima VSR.

Calcolo l'insieme delle relazioni LEGGE_DA e l'insieme delle SCRITTURE_FINALI di S_2 .

$$\text{LEGGE_DA}(S_2) = \{(r_2(x), w_3(x))\}$$

$\text{SCRITTURE_FINALI}(S_2) = \{w_3(x), w_1(y)\}$

Transazioni di S_2 :

$$\begin{aligned} T_1 &= r_1(x), w_1(x), w_1(y) \\ T_2 &= r_2(y), r_2(x) \\ T_3 &= w_3(x), r_3(y), w_3(y) \end{aligned}$$

Esiste una permutazione delle transazioni (tra le 6 possibili) che rappresenta uno schedule seriale view-equivalente a S_2 ? Vale a dire uno schedule seriale con lo stesso insieme di relazioni LEGGE_DA e SCRITTURE_FINALI ?

Quali permutazioni possiamo scartare a priori considerando l'insieme $\text{LEGGE_DA}(S_2)$?

Sicuramente per poter ottenere la relazione: $(r_2(x), w_3(x))$ è necessario considerare permutazioni dove T_3 precede T_2 .

Inoltre, per evitare che si generino altre relazioni LEGGE_DA è necessario che:

- T_1 precede T_3 altrimenti si genera la relazione: $(r_1(x), w_3(x))$
- T_3 precede T_1 altrimenti si genera la relazione: $(r_3(y), w_1(y))$

Questo è già sufficiente per concludere che non esistono schedule seriali view-equivalenti a S_2 e che quindi S_2 non è serializzabile (S_2 è nonSR).

Tuttavia, considerando anche l'insieme delle SCRITTURE_FINALI è necessario che:

- T_1 precede T_3 per conservare la scrittura finale: $w_3(x)$
- T_3 precede T_1 per conservare la scrittura finale: $w_1(y)$

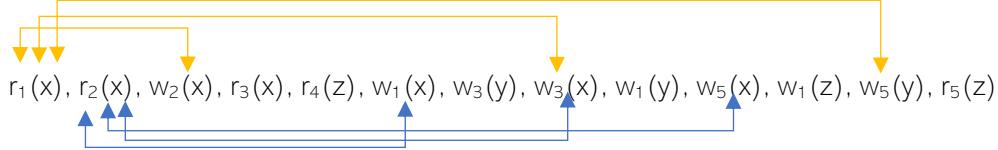
Quindi anche considerando solo le scritture finali ottengo lo stesso risultato.

SOLUZIONI

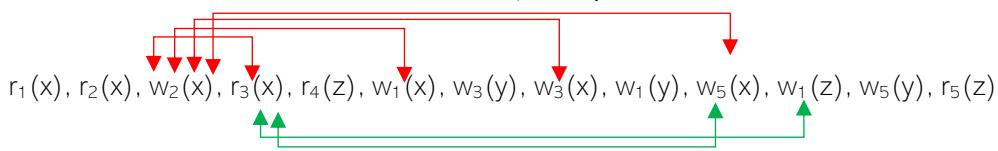
$$S_3 = r_1(x), r_2(x), w_2(x), r_3(x), r_4(z), w_1(x), w_3(y), w_3(x), w_1(y), w_5(x), w_1(z), w_5(y), r_5(z)$$

Scelgo di verificare prima CSR.

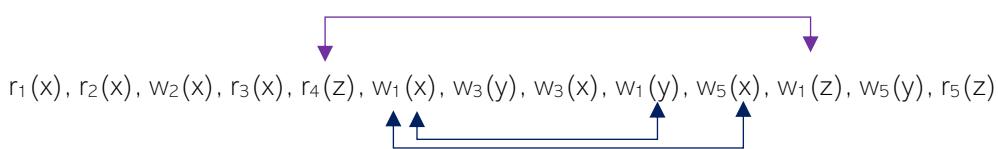
Calcolo l'insieme dei conflitti di S_3 .



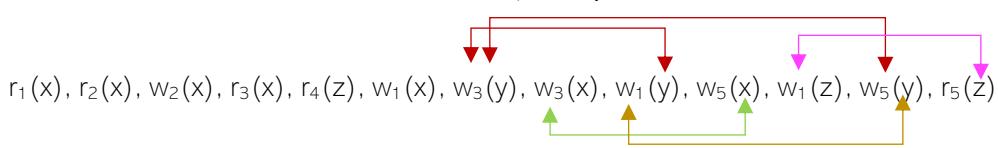
$$\text{Conflitti}(S_3) = \{(r_1(x), w_2(x)), (r_1(x), w_3(x)), (r_1(x), w_5(x)), \\ (r_2(x), w_1(x)), (r_2(x), w_3(x)), (r_2(x), w_5(x)), \\ \dots \text{da completare}\}$$



$$\text{Conflitti}(S_3) = \{(r_1(x), w_2(x)), (r_1(x), w_3(x)), (r_1(x), w_5(x)), \\ (r_2(x), w_1(x)), (r_2(x), w_3(x)), (r_2(x), w_5(x)), \\ (w_2(x), r_3(x)), (w_2(x), w_1(x)), (w_2(x), w_3(x)), (w_2(x), w_5(x)), \\ (r_3(x), w_1(x)), (r_3(x), w_5(x)), \\ \dots \text{da completare}\}$$

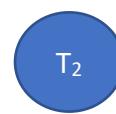


$$\text{Conflitti}(S_3) = \{(r_1(x), w_2(x)), (r_1(x), w_3(x)), (r_1(x), w_5(x)), \\ (r_2(x), w_1(x)), (r_2(x), w_3(x)), (r_2(x), w_5(x)), \\ (w_2(x), r_3(x)), (w_2(x), w_1(x)), (w_2(x), w_3(x)), (w_2(x), w_5(x)), \\ (r_3(x), w_1(x)), (r_3(x), w_5(x)), \\ (r_4(z), w_1(z)), (w_1(x), w_3(x)), (w_1(x), w_5(x)), \\ \dots \text{da completare}\}$$



$$\text{Conflitti}(S_3) = \{(r_1(x), w_2(x)), (r_1(x), w_3(x)), (r_1(x), w_5(x)), \\ (r_2(x), w_1(x)), (r_2(x), w_3(x)), (r_2(x), w_5(x)), \\ (w_2(x), r_3(x)), (w_2(x), w_1(x)), (w_2(x), w_3(x)), (w_2(x), w_5(x)), \\ (r_3(x), w_1(x)), (r_3(x), w_5(x)), \\ (r_4(z), w_1(z)), (w_1(x), w_3(x)), (w_1(x), w_5(x)), \\ (w_3(y), w_1(y)), (w_3(y), w_5(y)), (w_3(x), w_5(x)), (w_1(y), w_5(y)), (w_1(z), r_5(z))\}$$

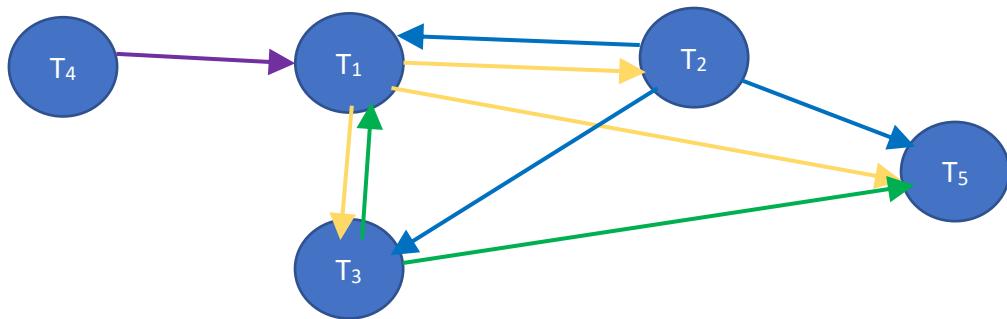
A questo punto genero il grafo dei conflitti. Ci sono in totale 5 transazioni nello schedule, pertanto avrò 5 nodi nel grafo.





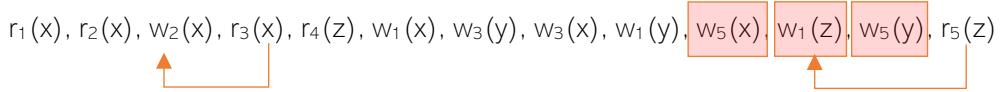
Per inserire gli archi consideriamo i conflitti calcolati in precedenza:

$$\text{Conflitti}(S_3) = \{(r_1(x), w_2(x)), (r_1(x), w_3(x)), (r_1(x), w_5(x)), \\ (r_2(x), w_1(x)), (r_2(x), w_3(x)), (r_2(x), w_5(x)), \\ (w_2(x), r_3(x)), (w_2(x), w_1(x)), (w_2(x), w_3(x)), (w_2(x), w_5(x)), \\ (r_3(x), w_1(x)), (r_3(x), w_5(x)), \\ (r_4(z), w_1(z)), (w_1(x), w_3(x)), (w_1(x), w_5(x)), \\ (w_3(y), w_1(y)), (w_3(y), w_5(y)), (w_3(x), w_5(x)), (w_1(y), w_5(y)), (w_1(z), r_5(z))\}$$



Poiché il grafo dei conflitti non è ACICLICO, lo schedule S_3 non è CSR.
Verifico se è VSR.

Calcolo l'insieme delle relazioni LEGGE_DA e l'insieme delle SCRITTURE_FINALI di S_3 .



$$\text{LEGGE_DA}(S_3) = \{(r_3(x), w_2(x)), (r_5(z), w_1(z))\}$$

$$\text{SCRITTURE_FINALI}(S_3) = \{w_5(x), w_1(z), w_5(y)\}$$

Transazioni:

$$t_1: r_1(x), w_1(x), w_1(y), w_1(z)$$

$$t_2: r_2(x), w_2(x)$$

$$t_3: r_3(x), w_3(y), w_3(x)$$

$$t_4: r_4(z)$$

$$t_5: w_5(x), w_5(y), r_5(z)$$

Considerando l'insieme SCRITTURE_FINALI(S_3) = { $w_5(x)$, $w_1(z)$, $w_5(y)$ }

Otteniamo che:

$$\text{Ultima scrittura su } y: w_5(y) \Rightarrow t_1 < t_5, t_3 < t_5$$

$$\text{Ultima scrittura su } z: w_5(x) \Rightarrow t_1 < t_5, t_2 < t_5, t_3 < t_5$$

Considerando l'insieme LEGGE_DA(S_3) = {($r_3(x), w_2(x)$), ($r_5(z), w_1(z)$)}.

Da cui otteniamo: $t_2 < t_3, t_1 < t_5$

A questo punto valutiamo le relazioni legge_da che potrebbero generarsi. Consideriamo le transazioni t_1 e t_2 .

$$t_1: r_1(x), w_1(x), w_1(y), w_1(z)$$

$$t_2: r_2(x), w_2(x)$$

Poiché è richiesto che $t_2 < t_3$, ipotizziamo di considerare il caso in cui anche $t_1 < t_3$.

Se $t_1 < t_2$ allora si genera la legge_da: $(r_2(x), w_1(x))$, relazione non presente nell'insieme LEGGE_DA(S_3).

Se invece $t_2 < t_1$ si genera la legge_da: $(r_1(x), w_2(x))$, relazione non presente nell'insieme LEGGE_DA(S_3). In quest'ultimo caso, fra l'altro si perde la relazione $(r_3(x), w_2(x))$.

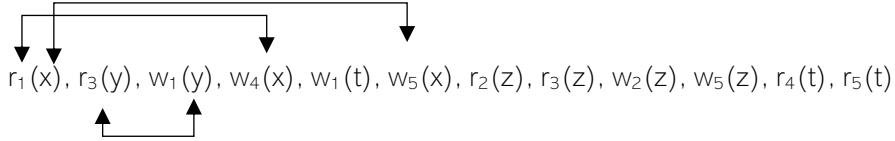
Poiché in qualsiasi permutazione o $t_1 < t_2$ o $t_2 < t_1$ nessun schedule seriale può presentare lo stesso insieme LEGGE_DA di S_3 . Quindi S_3 non è VSR. Pertanto è nonSR.

SOLUZIONI

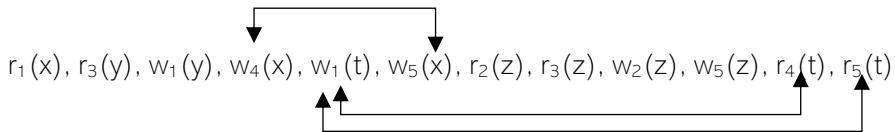
$$S_4 = r_1(x), r_3(y), w_1(y), w_4(x), w_1(t), w_5(x), r_2(z), r_3(z), w_2(z), w_5(z), r_4(t), r_5(t)$$

Scelgo di verificare prima CSR.

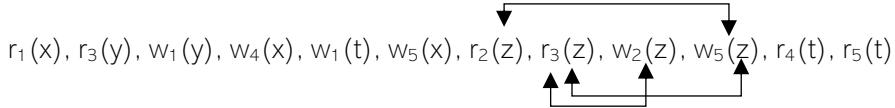
Calcolo l'insieme dei conflitti:



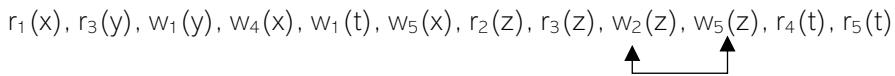
$$\text{Conflitti}(S_4) = \{(r_1(x), w_4(x)), (r_1(x), w_5(x)), (r_3(y), w_1(y)), \\ \dots \text{da completare}\}$$



$$\text{Conflitti}(S_4) = \{(r_1(x), w_4(x)), (r_1(x), w_5(x)), (r_3(y), w_1(y)), \\ (w_4(x), w_5(x)), (w_1(t), r_4(t)), (w_1(t), r_5(t)), \\ \dots \text{da completare}\}$$

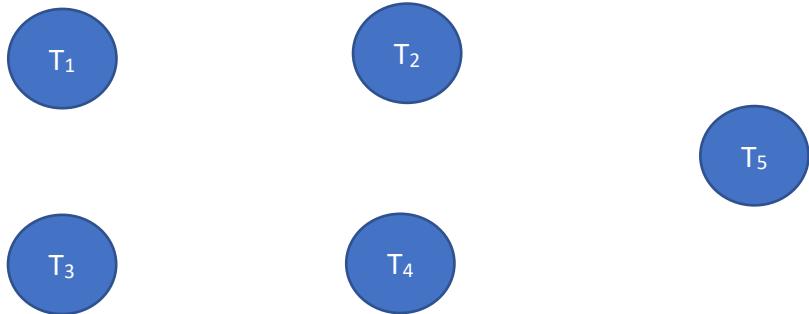


$$\text{Conflitti}(S_4) = \{(r_1(x), w_4(x)), (r_1(x), w_5(x)), (r_3(y), w_1(y)), \\ (w_4(x), w_5(x)), (w_1(t), r_4(t)), (w_1(t), r_5(t)), \\ (r_2(z), w_5(z)), (r_3(z), w_2(z)), (r_3(z), w_5(z)), \\ \dots \text{da completare}\}$$



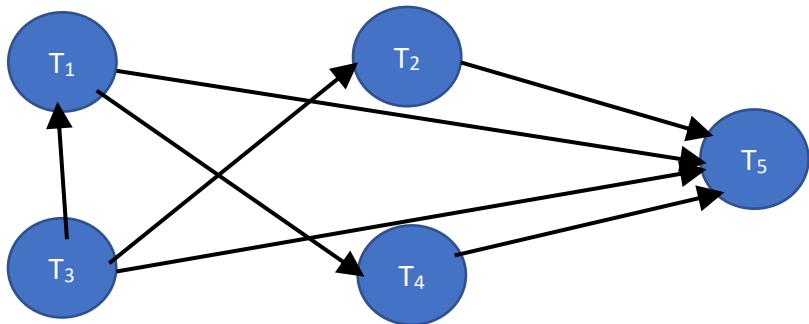
$$\text{Conflitti}(S_4) = \{(r_1(x), w_4(x)), (r_1(x), w_5(x)), (r_3(y), w_1(y)), \\ (w_4(x), w_5(x)), (w_1(t), r_4(t)), (w_1(t), r_5(t)), \\ (r_2(z), w_5(z)), (r_3(z), w_2(z)), (r_3(z), w_5(z)), \\ (w_2(z), w_5(z))\}$$

A questo punto genero il grafo dei conflitti. Ci sono in totale 5 transazioni nello schedule, pertanto avrò 5 nodi nel grafo.



Per inserire gli archi consideriamo i conflitti calcolati in precedenza:

$$\text{Conflitti}(S_4) = \{(r_1(x), w_4(x)), (r_1(x), w_5(x)), (r_3(y), w_1(y)), \\ (w_4(x), w_5(x)), (w_1(t), r_4(t)), (w_1(t), r_5(t)), \\ (r_2(z), w_5(z)), (r_3(z), w_2(z)), (r_3(z), w_5(z)), \\ (w_2(z), w_5(z))\}$$



Poiché il grafo dei conflitti è ACICLICO, lo schedule S_4 è CSR e quindi anche VSR.

Calcolo gli schedule seriali equivalenti:

$$r_1(x), r_3(y), w_1(y), w_4(x), w_1(t), w_5(x), r_2(z), r_3(z), w_2(z), w_5(z), r_4(t), r_5(t)$$

Transazioni:

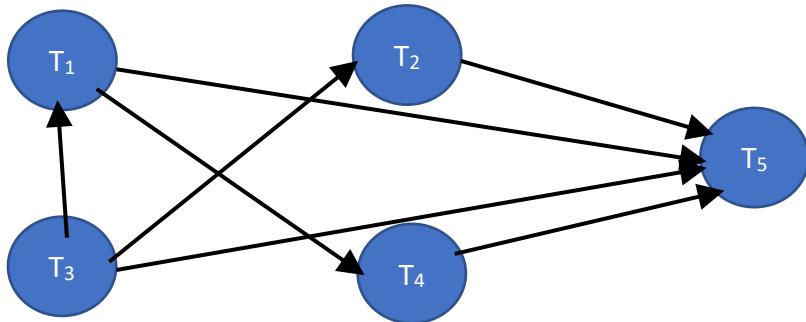
$$t_1: r_1(x), w_1(y), w_1(t)$$

$$t_2: r_2(z), w_2(z)$$

$$t_3: r_3(y), r_3(z)$$

$$t_4: w_4(x), r_4(t)$$

$$t_5: w_5(x), w_5(z), r_5(t)$$



Ordinamenti topologici desumibili dal grafo:

$$\text{ORD_TOP}_1: t_3, t_1, t_2, t_4, t_5$$

$$\text{ORD_TOP}_2: t_3, t_2, t_1, t_4, t_5$$

$$\text{ORD_TOP}_3: t_3, t_1, t_4, t_2, t_5$$

$$SS_1: r_3(y), r_3(z), r_1(x), w_1(y), w_1(t), r_2(z), w_2(z), w_4(x), r_4(t), w_5(x), w_5(z), r_5(t)$$

$$SS_2: r_3(y), r_3(z), r_2(z), w_2(z), r_1(x), w_1(y), w_1(t), w_4(x), r_4(t), w_5(x), w_5(z), r_5(t)$$

$$SS_3: r_3(y), r_3(z), r_1(x), w_1(y), w_1(t), w_4(x), r_4(t), r_2(z), w_2(z), w_5(x), w_5(z), r_5(t)$$