

ESERCIZI SUL TEST VSR (View-serializzabilità)

Verifichiamo che il test VSR funzioni per le anomalie di esecuzione concorrente.

PERDITA DI AGGIORNAMENTO

Questa anomalia può essere così descritta. Date due transazioni T_1 e T_2 di seguito descritte

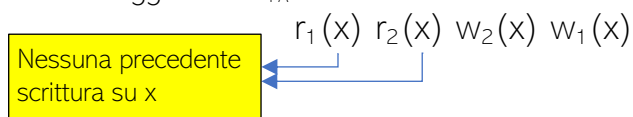
$T_1: r_1(x) \ w_1(x)$ $T_2: r_2(x) \ w_2(x)$

Lo schedule che rappresenta l'anomalia è il seguente

$$S_{PA} = r_1(x) \ r_2(x) \ w_2(x) \ w_1(x)$$

Ora per il test VSR è necessario innanzitutto caratterizzare S_{PA} calcolando l'insieme delle relazioni LeggeDa e l'insieme delle ScrittureFinali:

Calcolo delle relazioni LeggeDa di S_{PA}



per ogni operazione di lettura cerchiamo una precedente scrittura sulla stessa risorsa fatta da un'altra transazione. In questo caso si ottiene:

$$\text{LeggeDa}(S_{PA}) = \emptyset$$

Calcolo delle ScrittureFinali di S_{PA}

$$r_1(x) \ r_2(x) \ w_2(x) \ w_1(x)$$

Per ogni risorsa indicata nello schedule specificare l'ultima scrittura eseguita.

RISORSA	Ultima scrittura
x	$w_1(x)$

$$\text{ScrittureFinali}(S_{PA}) = \{w_1(x)\}$$

Quindi complessivamente per S_{PA} abbiamo:

$$\text{LeggeDa}(S_{PA}) = \emptyset$$

$$\text{ScrittureFinali}(S_{PA}) = \{w_1(x)\}$$

Ora è necessario generare tutti i possibili schedule seriali che eseguono le due transazioni. Tali schedule si ottengono generando tutte le possibili permutazioni dell'insieme di transazioni che partecipano allo schedule. Nel nostro caso sono solo due i possibili schedule seriali.

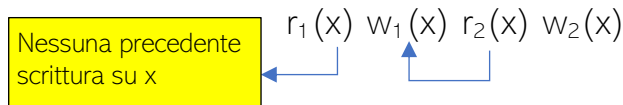
$$S_1 = r_1(x) \ w_1(x) \ r_2(x) \ w_2(x) \text{ che corrisponde alla permutazione } T_1, T_2$$

e

$$S_2 = r_2(x) \ w_2(x) \ r_1(x) \ w_1(x) \text{ che corrisponde alla permutazione } T_2, T_1$$

Verifichiamo ora se almeno uno dei due schedule seriali è view-equivalente a S_{PA} . Inizio da S_1 .

SCHEDULE S_1
 LeggeDa(S_1)



Quindi:

$$\text{LeggeDa}(S_1) = \{(r_2(x), w_1(x))\}$$

ScrittureFinali(S_1)

$r_1(x) \ w_1(x) \ r_2(x) \ w_2(x)$

RISORSA	Ultima scrittura
x	$w_2(x)$

$$\text{ScrittureFinali}(S_1) = \{w_2(x)\}$$

Quindi complessivamente per S_1 abbiamo:

$$\text{LeggeDa}(S_1) = \{(r_2(x), w_1(x))\}$$

$$\text{ScrittureFinali}(S_1) = \{w_2(x)\}$$

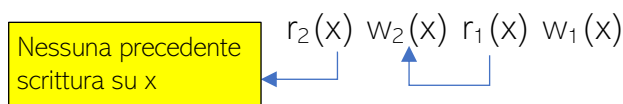
Se confrontiamo con S_{PA} possiamo notare che i due schedule hanno sia l'insieme LeggeDa() sia l'insieme ScrittureFinali() diversi e quindi possiamo concludere che:

$$S_{PA} \not\sim S_1$$

(S_{PA} non è view-equivalente a S_1)

Pertanto, dobbiamo procedere con S_2 .

SCHEDULE S_2
 LeggeDa(S_2)



Quindi:

$$\text{LeggeDa}(S_2) = \{(r_1(x), w_2(x))\}$$

ScrittureFinali(S_2)

$r_2(x) \ w_2(x) \ r_1(x) \ w_1(x)$

RISORSA	Ultima scrittura
x	$w_1(x)$

$$\text{ScrittureFinali}(S_2) = \{w_1(x)\}$$

Quindi complessivamente per S_2 abbiamo:

$$\text{LeggeDa}(S_2) = \{(r_1(x), w_2(x))\}$$

$$\text{ScrittureFinali}(S_2) = \{w_1(x)\}$$

Se confrontiamo con S_{PA} possiamo notare che i due schedule hanno l'insieme ScrittureFinali() uguale ma l'insieme LeggeDa() diverso e quindi possiamo concludere che:

$$S_{PA} \not\sim_V S_2$$

(S_{PA} non è view-equivalente a S_2)

Ora non essendovi altri schedule seriali possibili concludiamo che:

$$S_{PA} \text{ non è VSR}$$

(S_{PA} non è serializzabile)

LETTURA INCONSISTENTE

Questa anomalia può essere così descritta. Date due transazioni T_1 e T_2 di seguito descritte

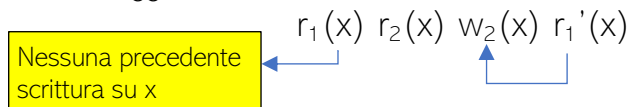
$T_1: r_1(x) \ r_1'(x) \quad T_2: r_2(x) \ w_2(x)$

Lo schedule che rappresenta l'anomalia è il seguente

$$S_{LI} = r_1(x) \ r_2(x) \ w_2(x) \ r_1'(x)$$

Ora per il test VSR è necessario innanzitutto caratterizzare S_{LI} calcolando l'insieme delle relazioni LeggeDa e l'insieme delle ScrittureFinali:

Calcolo delle relazioni LeggeDa di S_{LI}



per ogni operazione di lettura cerchiamo una precedente scrittura sulla stessa risorsa fatta da un'altra transazione. In questo caso si ottiene:

$$\text{LeggeDa}(S_{LI}) = \{(r_1'(x), w_2(x))\}$$

Calcolo delle ScrittureFinali di S_{LI}

$$r_1(x) \ r_2(x) \ w_2(x) \ r_1'(x)$$

Per ogni risorsa indicata nello schedule specificare l'ultima scrittura eseguita.

RISORSA	Ultima scrittura
x	$w_2(x)$

$$\text{ScrittureFinali}(S_{LI}) = \{w_2(x)\}$$

Quindi complessivamente per S_{LI} abbiamo:

$$\text{LeggeDa}(S_{LI}) = \{(r_1'(x), w_2(x))\}$$

$$\text{ScrittureFinali}(S_{LI}) = \{w_2(x)\}$$

Ora è necessario generare tutti i possibili schedule seriali che eseguono le due transazioni. Tali schedule si ottengono generando tutte le possibili permutazioni dell'insieme di transazioni che partecipano allo schedule. Nel nostro caso sono solo due i possibili schedule seriali.

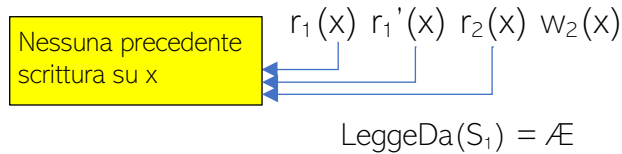
$$S_1 = r_1(x) \ r_1'(x) \ r_2(x) \ w_2(x) \text{ che corrisponde alla permutazione } T_1, T_2$$

e

$$S_2 = r_2(x) \ w_2(x) \ r_1(x) \ r_1'(x) \text{ che corrisponde alla permutazione } T_2, T_1$$

Verifichiamo ora se almeno uno dei due schedule seriali è view-equivalente a S_{LI} . Iniziamo da S_1 .

SCHEDULE S_1
 LeggeDa(S_1)



ScrittureFinali(S_1)

$r_1(x) \ r_1'(x) \ r_2(x) \ w_2(x)$

RISORSA	Ultima scrittura
x	$w_2(x)$

ScrittureFinali(S_1) = $\{w_2(x)\}$

Quindi complessivamente per S_1 abbiamo:

LeggeDa(S_1) = \mathcal{AE}

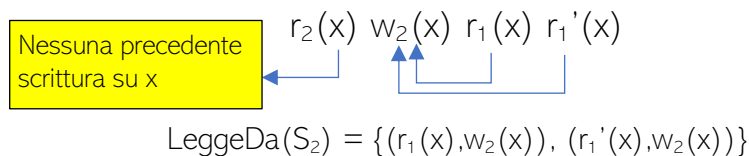
ScrittureFinali(S_1) = $\{w_2(x)\}$

Se confrontiamo con S_{LI} possiamo notare che i due schedule hanno l'insieme ScrittureFinali() uguale mentre l'insieme LeggeDa() diverso e quindi possiamo concludere che:

$S_{LI} \not\sim S_1$
 (S_{LI} non è view-equivalente a S_1)

Pertanto, dobbiamo procedere con S_2 .

SCHEDULE S_2
 LeggeDa(S_2)



ScrittureFinali(S_2)

$r_2(x) \ w_2(x) \ r_1(x) \ r_1'(x)$

RISORSA	Ultima scrittura
x	$w_2(x)$

ScrittureFinali(S_2) = $\{w_2(x)\}$

Quindi complessivamente per S_2 abbiamo:

LeggeDa(S_2) = $\{(r_1(x), w_2(x)), (r_1'(x), w_2(x))\}$

ScrittureFinali(S_2) = $\{w_2(x)\}$

Se confrontiamo con S_{LI} possiamo notare che i due schedule hanno l'insieme ScrittureFinali() uguale ma l'insieme LeggeDa() diverso e quindi possiamo concludere che:

$S_{LI} \not\sim S_2$

$(S_{L1}$ non è view-equivalente a S_2)

Ora non essendovi altri schedule seriali possibili concludiamo che:

S_{L1} non è VSR
(S_{L1} non è serializzabile)