

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

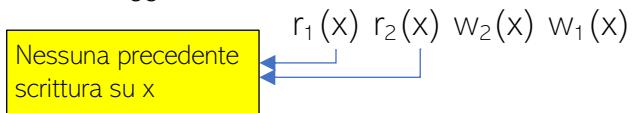
$T_1: r_1(x) \ w_1(x)$ $T_2: r_2(x) \ w_2(x)$

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:

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

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

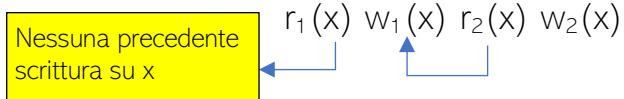
Ora è necessario generare tutti i possibili schedule seriali che eseguono le due transazioni. Tali schedule si ottengono generando tutte le possibili permutazioni dell'insieme di transazioni che partecipano allo schedule. Nel nostro caso sono solo due i possibili schedule seriali.

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

$S_2 = r_2(x) \ w_2(x) \ r_1(x) \ w_1(x)$ 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 .

SCHEDELE S_1
 $\text{LeggeDa}(S_1)$



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:

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

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

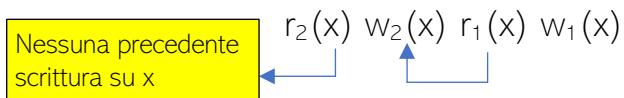
Se confrontiamo con S_{PA} possiamo notare che i due schedule hanno sia l'insieme $\text{LeggeDa}()$ sia l'insieme $\text{ScrittureFinali}()$ diversi e quindi possiamo concludere che:

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

(S_{PA} non è view-equivalente a S_1)

Pertanto, dobbiamo procedere con S_2 .

SCHEDELE S_2
 $\text{LeggeDa}(S_2)$



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:

$S_{PA} \not\sim S_2$
(S_{PA} non è view-equivalente a S_2)

Ora non essendovi altri schedule seriali possibili concludiamo che:

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

LETTURA INCONSENTE

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

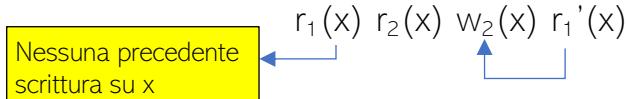
$T_1: r_1(x) \ r_1'(x)$ $T_2: r_2(x) \ w_2(x)$

Lo schedule che rappresenta l'anomalia è il seguente

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

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

Calcolo delle relazioni LeggeDa di S_{\sqcup}



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

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

Calcolo delle ScrittureFinali di S_{\sqcup}

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

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

RISORSA	Ultima scrittura
x	w ₂ (x)

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

Quindi complessivamente per S_{\sqcup} abbiamo:

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

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

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.

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

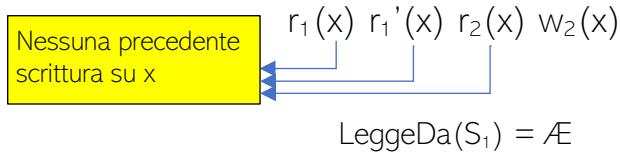
e

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

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

SCHEDE S₁

LeggeDa(S₁)



ScrittureFinali(S₁)

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

RISORSA	Ultima scrittura
x	$w_2(x)$

ScrittureFinali(S₁) = { $w_2(x)$ }

Quindi complessivamente per S₁ abbiamo:

LeggeDa(S₁) = A&E

ScrittureFinali(S₁) = { $w_2(x)$ }

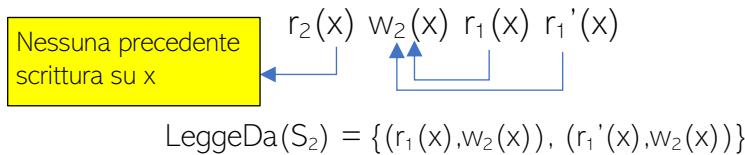
Se confrontiamo con S_U possiamo notare che i due schedule hanno l'insieme ScrittureFinali() uguale mentre l'insieme LeggeDa() diverso e quindi possiamo concludere che:

$S_{U} \not\propto S_1$
(S_{U} non è view-equivalente a S_1)

Pertanto, dobbiamo procedere con S₂.

SCHEDE S₂

LeggeDa(S₂)



ScrittureFinali(S₂)

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

RISORSA	Ultima scrittura
x	$w_2(x)$

ScrittureFinali(S₂) = { $w_2(x)$ }

Quindi complessivamente per S₂ abbiamo:

LeggeDa(S₂) = {(r₁(x),w₂(x)), (r₁'(x),w₂(x))}

ScrittureFinali(S₂) = { $w_2(x)$ }

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

$S_{U} \not\propto S_2$

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

Ora non essendovi altri schedule seriali possibili concludiamo che:

S_{\sqcup} non è VSR
 $(S_{\sqcup}$ non è serializzabile)