

# Lezione 28– Write-only, read-only

Prof.ssa Maria De Marsico  
demarsico@di.uniroma1.it



SAPIENZA  
UNIVERSITÀ DI ROMA



- Una transazione può accedere ad un item solo per leggerlo, senza modificarlo
- Se una transazione desidera **solo leggere** un item  $X$  effettua una  $rlock(X)$  che impedisce a qualsiasi altra transazione di modificare  $X$ , ma non di leggere  $X$
- Se una transazione desidera **solo modificare** un item  $X$  effettua un  $wlock(X)$ ; in tal caso nessuna altra transazione può leggere o modificare  $X$
- Entrambi i **lock** sono rilasciati mediante una  $unlock(X)$

- Una transazione è una sequenza di operazioni di *rlock*, *wlock* e *unlock*
- ogni *rlock*(X) implica la **lettura** di X
- ogni *unlock*(X) associato a una *wlock*(X) implica la **scrittura** di X
- a differenza del modello precedente una *wlock*(X) **non** implica anche una **lettura**
- non c'è necessariamente una relazione di **contenimento** tra l'insieme degli item **letti** (**read set**) e quello degli item **scritti** (**write set**) da una transazione.

$T_1$
$rlock(X)$
$unlock(X)$
$wlock(Y)$
$unlock(Y) f_1(X)$

Il nuovo valore viene calcolato da una funzione che è associata in modo univoco ad ogni coppia wlock-unlock ed ha per **argomenti** tutti gli item **letti** (**rlocked**) dalla transazione **prima** dell'operazione di unlock

**Mancano** dalla funzione gli item **wlocked**, perché il loro valore **non è stato letto e quindi non influenza la scrittura successiva**

Due schedule sono *equivalenti* se

- producono lo **stesso valore** per **ogni item** su cui **viene effettuato un wlock** (le **formule** che danno i valori finali per ciascun item sono le **stesse**)
- **NOTA non si assume** che il valore di un item **letto** da una transazione **sia significativo** indipendentemente dal fatto che abbia influenza sul valore finale di qualche item prodotto dalla transazione.

Partiamo da una definizione simile a quella data per i modelli precedenti.

Uno schedule è **serializzabile** se esiste uno schedule **seriale** che **produce** lo **stesso valore per ogni item** su cui lo schedule effettua operazioni di **scrittura**

Consideriamo il seguente schedule di tre transazioni:

$T_1$	$T_2$	$T_3$
$wlock(A)$ $unlock(A)$	$wlock(C)$ $unlock(C)$ $rlock(A)$ $wlock(B)$ $unlock(A)$ $unlock(B)$	
$rlock(C)$ $wlock(D)$ $unlock(C)$ $unlock(D)$		
		$wlock(B)$ $wlock(D)$ $unlock(B)$ $unlock(D)$

$$A=f_1( )$$

$$C=f_2( )$$

$$B=f_3(f_1( ))$$

$$D=f_4(f_2( ))$$

$$B=f_5( )$$

$$D=f_6( )$$

I valori finali degli  
item sono  
evidenziati in rosso...

... e coincidono con i valori finali degli item  
prodotti dallo schedule seriale:

$T_1$	$T_2$	$T_3$
$wlock(A)$ $unlock(A)$ $rlock(C)$ $wlock(D)$ $unlock(C)$ $unlock(D)$	$wlock(C)$ $unlock(C)$ $rlock(A)$ $wlock(B)$ $unlock(A)$ $unlock(B)$	$wlock(B)$ $wlock(D)$ $unlock(B)$ $unlock(D)$

$$A = f_1( )$$

$$D = f_4(C_0)$$

$$C = f_2( )$$

$$B = f_3(f_1( ))$$

$$B = f_5( )$$

$$D = f_6( )$$

Quindi in base alla  
definizione  
di serializzabilità data,  
lo schedule è serializzabile!



$T_1$	$T_2$	$T_3$
$wlock(A)$ $unlock(A)$	$wlock(C)$ $unlock(C)$ $rlock(A)$ $wlock(B)$ $unlock(A)$ $unlock(B)$	
$rlock(C)$ $wlock(D)$ $unlock(C)$ $unlock(D)$		$wlock(B)$ $wlock(D)$ $unlock(B)$ $unlock(D)$

$$A = f_1( )$$

$$C = f_2( )$$

$$B = f_3(f_1( ))$$

$$D = f_4(f_2( ))$$

Ma se  $T_3$  non viene  
 eseguita i valori  
 finali degli item

...

$T_1$	$T_2$	$T_3$
$wlock(A)$ $unlock(A)$ $rlock(C)$ $wlock(D)$ $unlock(C)$ $unlock(D)$	$wlock(C)$ $unlock(C)$ $rlock(A)$ $wlock(B)$ $unlock(A)$ $unlock(B)$	$wlock(B)$ $wlock(D)$ $unlock(B)$ $unlock(D)$

$$A = f_1( )$$

$$D = f_4(C_0)$$

$$C = f_2( )$$

$$B = f_3(f_1( ))$$

...non coincidono nè con  
quelli prodotti dallo  
schedule seriale  $T_1 T_2$   
...

$T_1$	$T_2$	$T_3$
$wlock(A)$ $unlock(A)$ $rlock(C)$ $wlock(D)$ $unlock(C)$ $unlock(D)$	$wlock(C)$ $unlock(C)$ $rlock(A)$ $wlock(B)$ $unlock(A)$ $unlock(B)$	$wlock(B)$ $wlock(D)$ $unlock(B)$ $unlock(D)$

$$C = f_2( )$$

$$B = f_3(A_0)$$

$$A = f_1( )$$

$$D = f_4(f_2( ))$$

...nè con quelli prodotti  
dallo schedule seriale  $T_2 T_1$

e quindi **non** è  
**serializzabile!!!**



Occorre una diversa definizione di equivalenza di schedule e quindi una diversa definizione di serializzabilità

Una schedule seriale  $S'$  è **equivalente** ad una schedule  $S$  se soddisfa i seguenti vincoli:

1. se in  $S$  una transazione  $T_2$  **legge** il valore di un item  $X$  **scritto** da una transazione  $T_1$  allora in  $S'$   $T_1$  **deve precedere**  $T_2$
2. se  $T_3$  è una terza transazione che **scrive**  $X$  allora in  $S'$   $T_3$  **deve precedere**  $T_1$  o **seguire**  $T_2$

Inoltre

3. se una transazione **legge** in  $S$  il valore **iniziale** di un item  $X$  allora in  $S'$  **deve precedere qualsiasi transazione che scriva  $X$**
4. se una transazione **scrive** il valore **finale** di un item  $X$  allora in  $S'$  **deve seguire qualsiasi transazione che scriva  $X$ .**

# Serializzabilità (conflict serializability)



SAPIENZA  
UNIVERSITÀ DI ROMA  
DIPARTIMENTO DI INFORMATICA

I vincoli 3 e 4 sono riassorbiti dai vincoli 1 e 2 se postuliamo l'esistenza di

una transazione **iniziale**  $T_0$  che **scrive i valori iniziali** di **tutti** gli item (senza **leggerne nessuno**) e

una transazione **finale**  $T_f$  che **legge i valori finali** di tutti gli item (senza **scriverne nessuno**)

# Testare la serializzabilità



SAPIENZA  
UNIVERSITÀ DI ROMA  
DIPARTIMENTO DI INFORMATICA

## Algoritmo 2

- Dato uno schedule  $S$

- **Passo 1**

- crea un **poligrafo diretto**  $P$

**nodi:** transazioni +  $T_0$  +  $T_f$

**archi**

- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item  $X$  **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché non c'è nessun cammino in  $P$  da  $T$  a  $T_f$ )



- vengono creati gli archi in **accordo al vincolo 2**: per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

- l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$
- l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$
- la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$

# Testare la serializzabilità



SAPIENZA  
UNIVERSITÀ DI ROMA  
DIPARTIMENTO DI INFORMATICA

- *Passo 2*
- Se almeno uno dei grafi ottenuti prendendo un solo arco da ogni coppia di archi alternativi è **aciclico** allora lo schedule è **serializzabile** e
- uno schedule seriale **equivalente** si ottiene mediante **sorting topologico** del grafo di serializzazione

	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18				$wlock(A)$
19		$unlock(B)$		
20				$wlock(B)$
21				$unlock(B)$
22		$unlock(D)$		
23				$unlock(C)$
24				$unlock(A)$

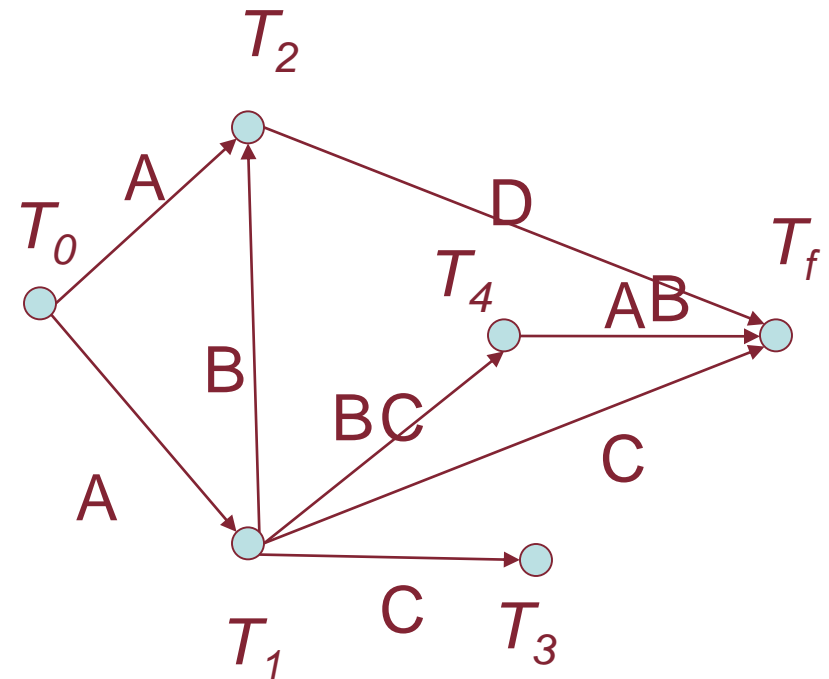
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18			$wlock(A)$	
19		$unlock(B)$		
20			$wlock(B)$	
21			$unlock(B)$	
22		$unlock(D)$		
23				$unlock(C)$
24				$unlock(A)$

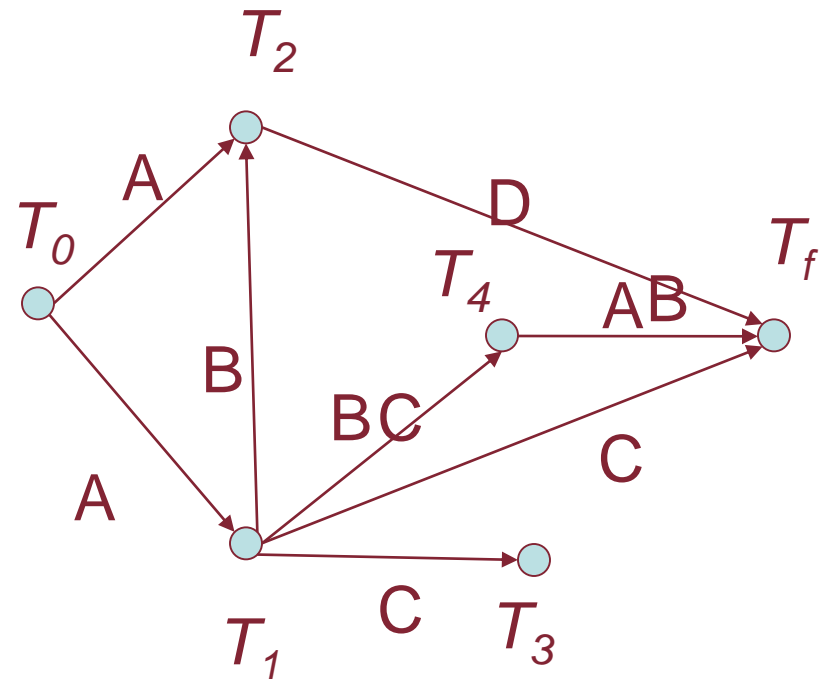
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		$unlock(B)$
14			$unlock(C)$	
15		$rlock(B)$	$unlock(A)$	
16				$wlock(A)$
17		$unlock(B)$		$wlock(B)$
18				$unlock(B)$
19		$unlock(D)$		
20				$unlock(C)$
21				$unlock(A)$
22				
23				
24				

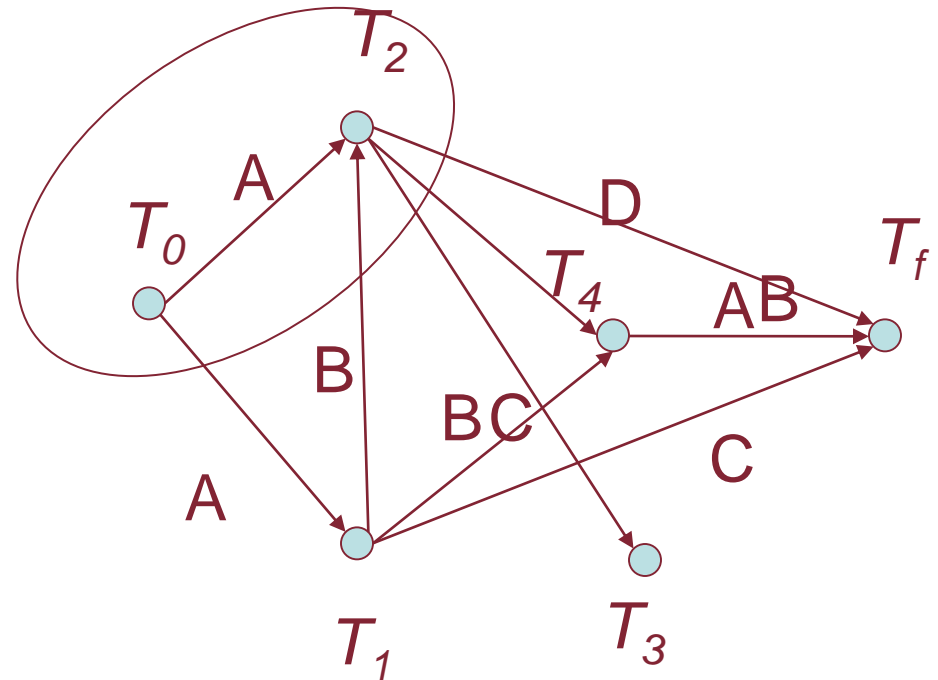
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18			$wlock(A)$	
19		$unlock(B)$		
20			$wlock(B)$	
21			$unlock(B)$	
22		$unlock(D)$		
23			$unlock(C)$	
24			$unlock(A)$	

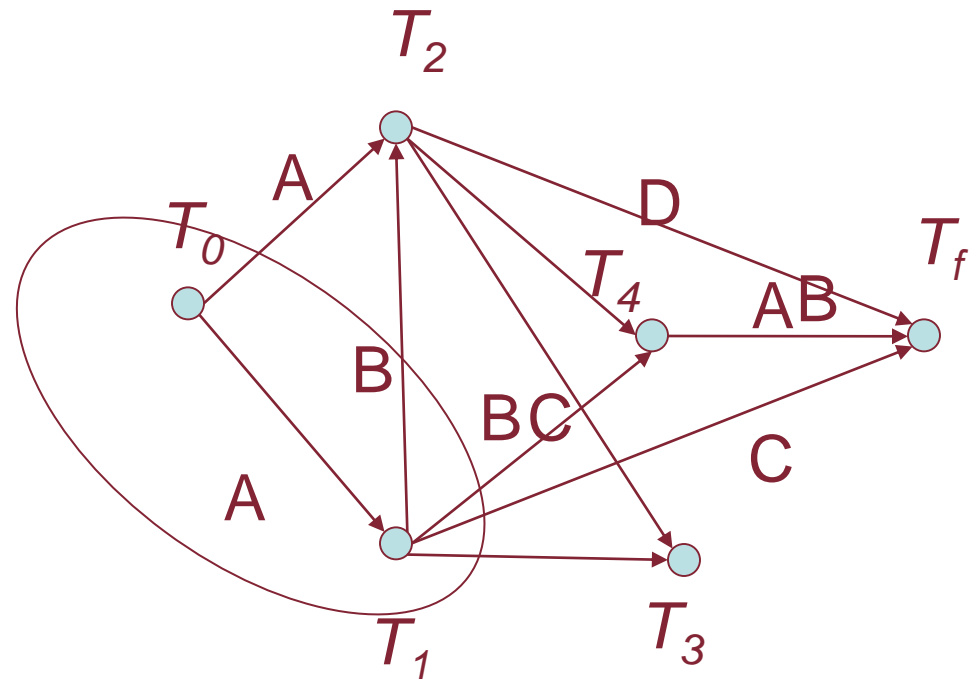
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



Dovremmo aggiungere T1-T4 ma c'è già anche se a causa di item diversi

	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18			$wlock(A)$	
19		$unlock(B)$		
20			$wlock(B)$	
21			$unlock(B)$	
22		$unlock(D)$		
23				$unlock(C)$
24				$unlock(A)$

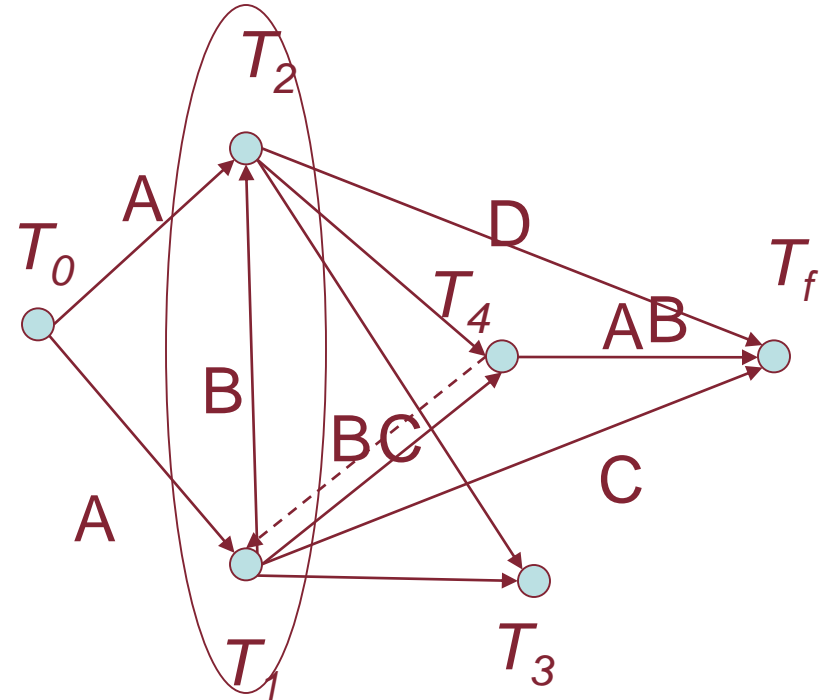
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18			$wlock(A)$	
19		$unlock(B)$		
20			$wlock(B)$	
21			$unlock(B)$	
22		$unlock(D)$		
23			$unlock(C)$	
24			$unlock(A)$	

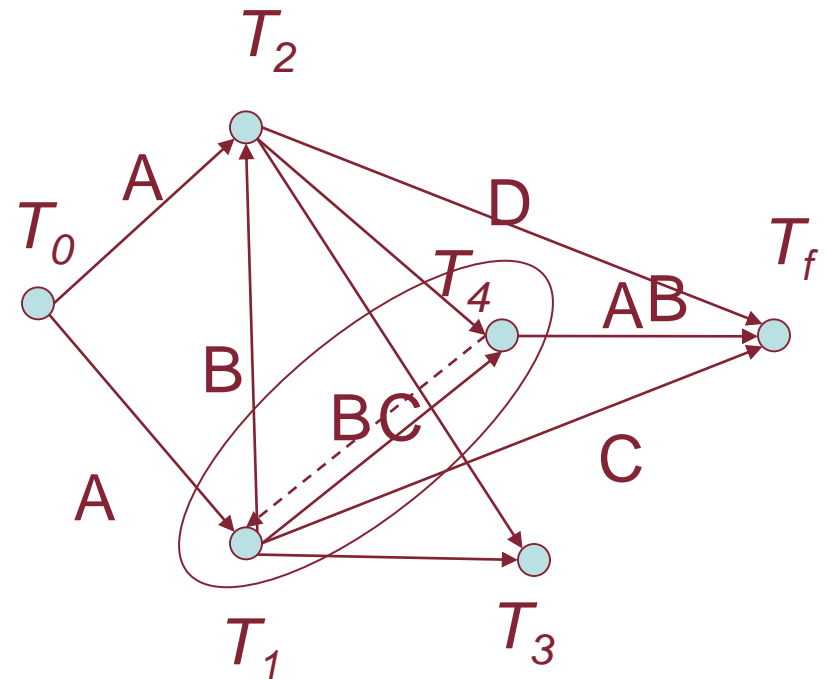
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$





	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18				$wlock(A)$
19		$unlock(B)$		
20				$wlock(B)$
21				$unlock(B)$
22		$unlock(D)$		
23				$unlock(C)$
24				$unlock(A)$

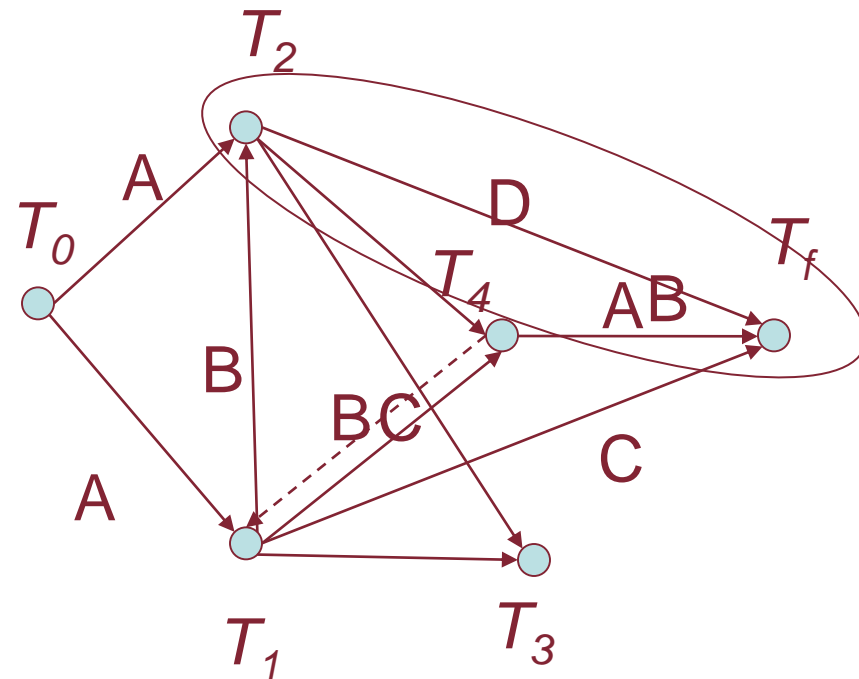
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18				$wlock(A)$
19		$unlock(B)$		
20				$wlock(B)$
21				$unlock(B)$
22		$unlock(D)$		
23				$unlock(C)$
24				$unlock(A)$

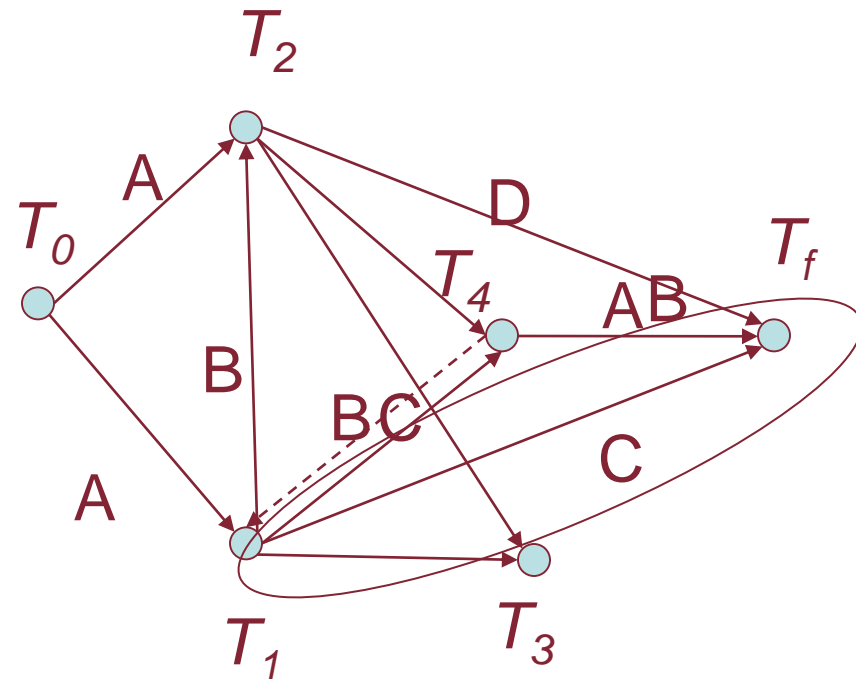
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

si aggiunge:

l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$



	$T_1$	$T_2$	$T_3$	$T_4$
1		$rlock(A)$		
2	$rlock(A)$			
3	$wlock(C)$			
4	$unlock(C)$			
5			$rlock(C)$	
6	$wlock(B)$			
7	$unlock(B)$			
8				$rlock(B)$
9	$unlock(A)$			
10		$unlock(A)$		
11			$wlock(A)$	
12				$rlock(C)$
13		$wlock(D)$		
14			$unlock(B)$	
15			$unlock(C)$	
16		$rlock(B)$		
17		$unlock(A)$		
18			$wlock(A)$	
19		$unlock(B)$		
20			$wlock(B)$	
21			$unlock(B)$	
22		$unlock(D)$		
23			$unlock(C)$	
24			$unlock(A)$	

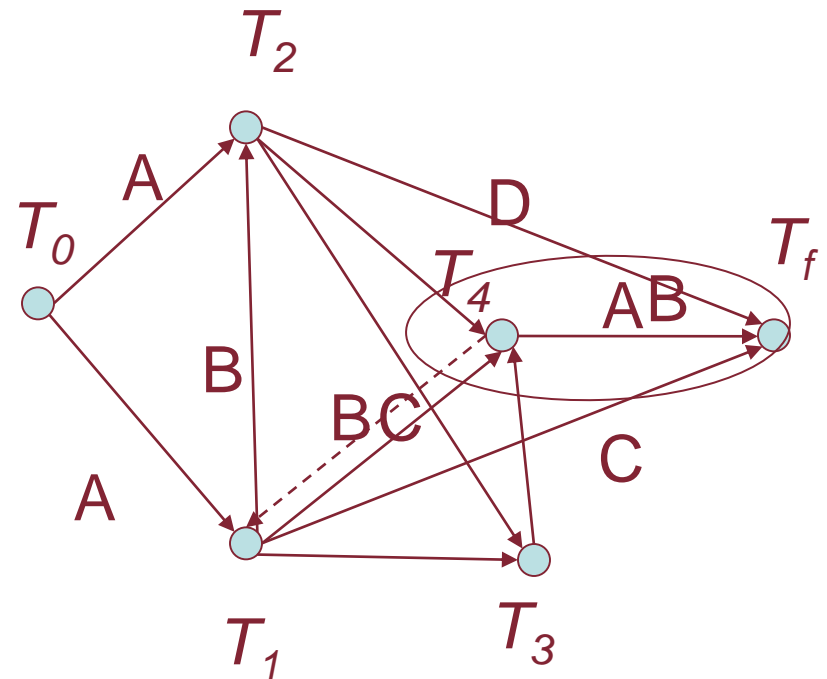
- vengono creati gli archi in accordo al **vincolo 1**: se una transazione  $T_2$  **legge** il valore di un item X **scritto** da una transazione  $T_1$  viene aggiunto l'arco  $T_1 \rightarrow T_2$
- vengono eliminati tutti gli archi **entranti** in **transazioni inutili** (una transazione inutile  $T$  può essere individuata facilmente perché **non c'è nessun cammino in  $P$  da  $T$  a  $T_f$** )
- per ogni arco  $T_1 \rightarrow T_2$  se  $T_3$  è transazione **distinta** da quella iniziale che **scrive** un item che ha **imposto** l'esistenza dell'arco  $T_1 \rightarrow T_2$ ,

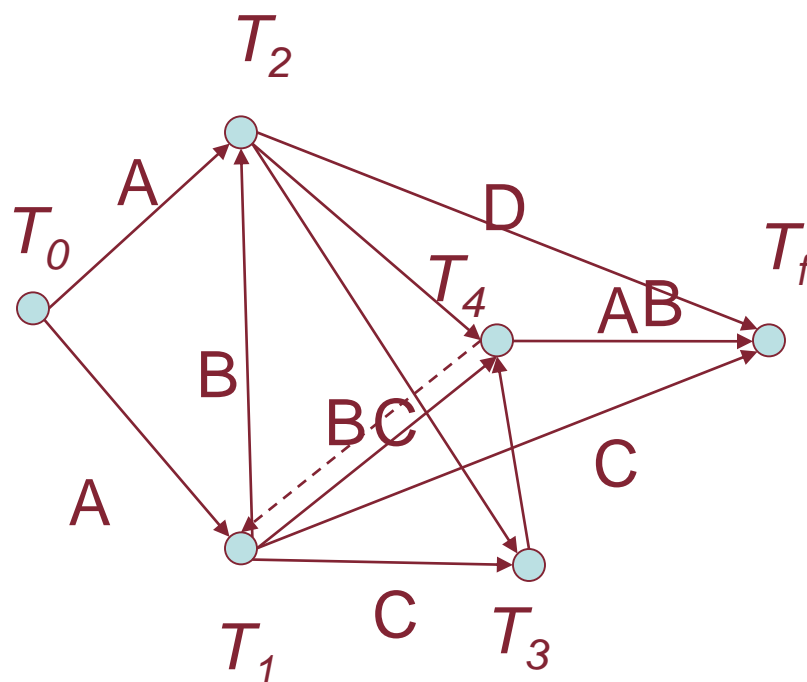
si aggiunge:

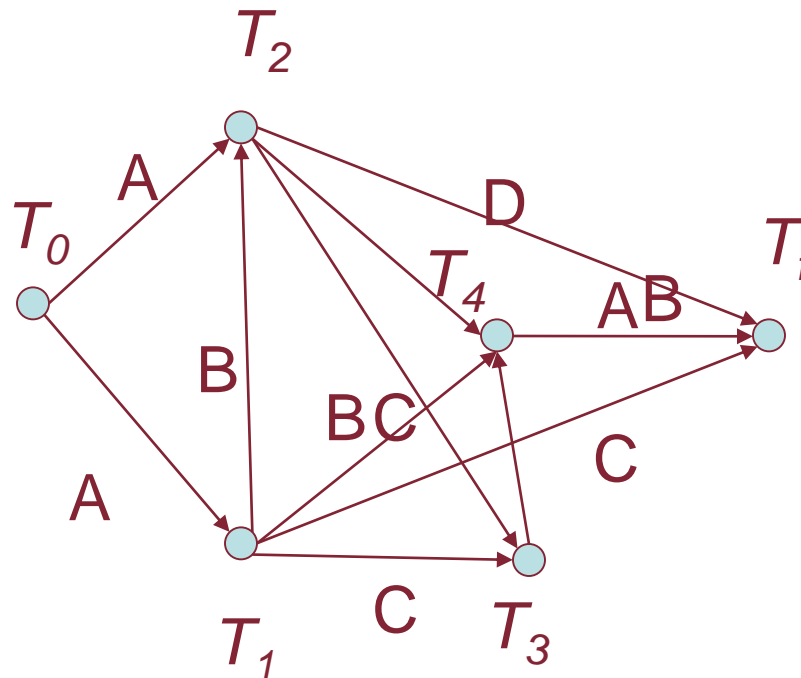
l'arco  $T_2 \rightarrow T_3$  se  $T_1 = T_0$  e  $T_2 \neq T_f$

l'arco  $T_3 \rightarrow T_1$  se  $T_2 = T_f$  e  $T_1 \neq T_0$

la **coppia di archi alternativi**  $T_3 \rightarrow T_1$  e  $T_2 \rightarrow T_3$  se  $T_1 \neq T_0$  e  $T_2 \neq T_f$





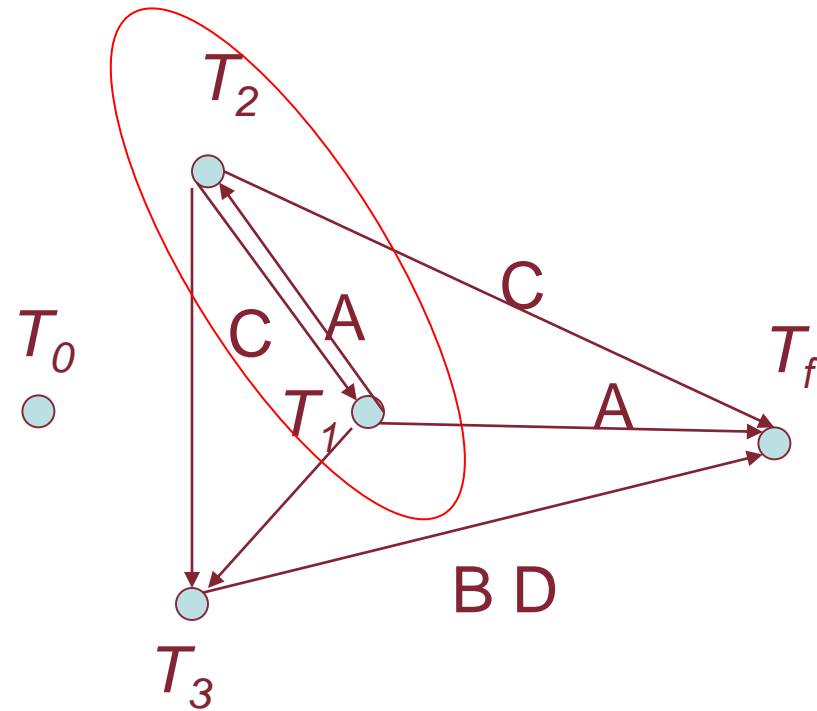


Lo schedule è serializzabile.

Uno schedule seriale equivalente è

$T_1 T_2 T_3 T_4$

$T_1$	$T_2$	$T_3$
$wlock(A)$ $unlock(A)$	$wlock(C)$ $unlock(C)$ $rlock(A)$ $wlock(B)$ $unlock(A)$ $unlock(B)$	
$rlock(C)$ $wlock(D)$ $unlock(C)$ $unlock(D)$		$wlock(B)$ $wlock(D)$ $unlock(B)$ $unlock(D)$



Lo schedule non è serializzabile