

# Serializability

S	
T <sub>1</sub>	T <sub>2</sub>
R(A) W(A)	
	R(A) W(A)



# Serializability

S	
T <sub>1</sub>	T <sub>2</sub>
R(A) W(A)	

↓

R(A)  
W(A)



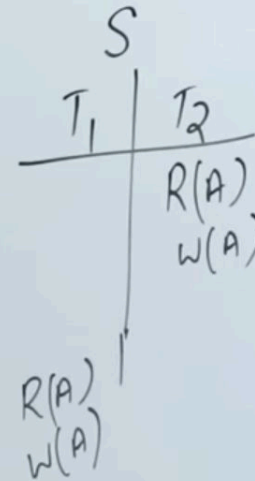
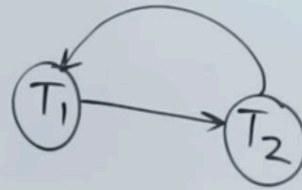
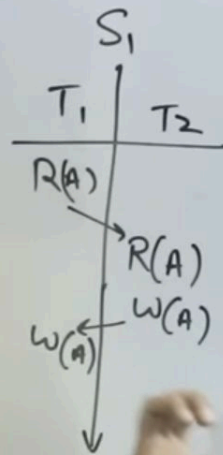
S	
T <sub>1</sub>	T <sub>2</sub>
	R(A) W(A)

↓

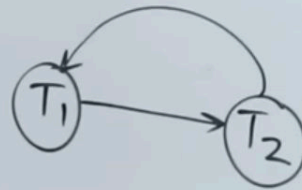
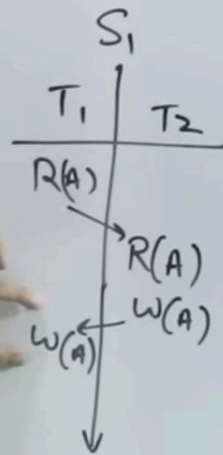
R(A)  
W(A)



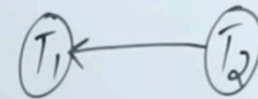
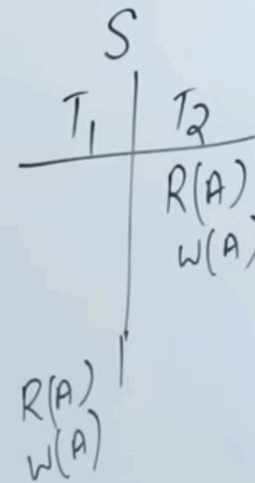
# Serializability



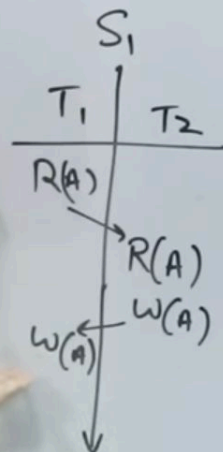
# Serializability



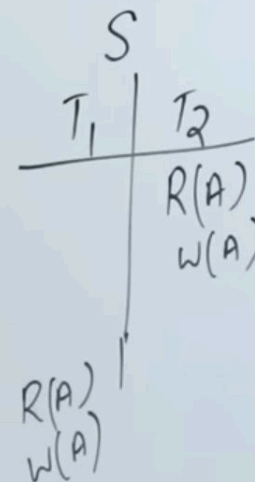
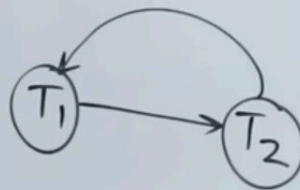
$T_1 \rightarrow T_2$   
OR  
 $T_2 \rightarrow T_1$



Serializability  $\begin{cases} \text{Conflict} \\ \text{View} \end{cases}$

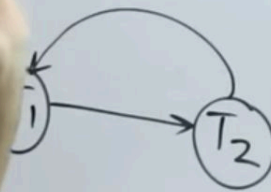


$T_1 \rightarrow T_2$   
OR  
 $T_2 \rightarrow T_1$



Serializability < Conflict View

$T_2$

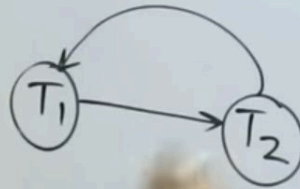


$T_2 \rightarrow T_1$

S		
$T_1$	$T_2$	$T_3$
	R(A)	
		R(A) W(A)
	W(A)	
R(B) W(B)		
		W(B)

Serializability  $\swarrow$  Conflict View

S
T <sub>1</sub>
R(A)



$\rightarrow T_2$

S	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
		R(A)	
			R(A)
			W(A)
		W(A)	
	R(B)		
	W(B)		
			W(B)

$T_1 \rightarrow T_2 \rightarrow T_3$   
 $T_1 \rightarrow T_3 \rightarrow T_2$   
 $T_2 \rightarrow T_3 \rightarrow T_1$   
 $T_2 \rightarrow T_1 \rightarrow T_3$   
 $T_3 \rightarrow T_1 \rightarrow T_2$   
 $T_3 \rightarrow T_2 \rightarrow T_1$



Serializability – To find an equivalent schedule of a parallel schedule which is serializable.



# Conflict Equivalent schedules

# "Conflict Equivalent"

Non Conflict Pairs

Conflict Pairs

dict

$$S \equiv S'$$

S	
T <sub>1</sub>	T <sub>2</sub>
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	

S'	
T <sub>1</sub>	T <sub>2</sub>
R(A)	
W(A)	
R(B)	
	R(A)
	W(A)

# "Conflict Equivalent"

R(A)	R(A)	Non Conflict Pairs
R(A)	W(A)	
W(A)	R(A)	Conflict Pairs
W(A)	W(A)	
R(B)	R(A)	Non Conflict
W(B)	R(A)	
R(B)	W(A)	
W(A)	W(B)	

$S \equiv S'$

S	
T <sub>1</sub>	T <sub>2</sub>
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	

S'	
T <sub>1</sub>	T <sub>2</sub>
R(A)	
W(A)	
R(B)	
	R(A)
	W(A)

# "Conflict Equivalent"

$$S \equiv S'$$

$R(A)$  } Non Conflict Pairs

$W(A)$   
 $R(A)$  } Conflict Pairs  
 $W(A)$

$R(A)$   
 $R(A)$  } Non Conflict  
 $W(A)$   
 $W(B)$

S	
T <sub>1</sub>	T <sub>2</sub>
R(A)	
W(A)	
	R(A)
	W(A)
	R(B)

S'	
T <sub>1</sub>	T <sub>2</sub>
R(A)	
W(A)	
R(B)	
	R(A)
	W(A)

Adjacent Non Conflict pairs

# "Conflict Equivalent"

$$S \equiv S'$$

$R(A)$   $R(A)$  } N  
 $R(A)$   $W(A)$  }  
 $W(A)$   $R(A)$  }  
 $W(A)$   $W(A)$  }  
 $R(B)$   $R(A)$  }  
 $W(B)$   $W(A)$  }  
 $R(B)$   $W(A)$  }  
 $W(A)$   $W(B)$  }

$T_1$	$T_2$
$R(A)$	
$W(A)$	
$R(B)$	$R(A)$
	$W(A)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
	$R(A)$
$R(B)$	$W(A)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
$R(B)$	
	$R(A)$
	$W(A)$

Adjacent Non Conflict pairs

"Conflict Equivalent"

$S \equiv S'$

$R(A) \in N$

$W(A)$

$R(A)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
	$R(A)$
	$R(B)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
$R(B)$	
	$R(A)$
	$W(A)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
$R(B)$	
	$R(A)$
	$W(A)$

Adjacent Non Conflict pairs



"Conflict Equivalent"

$$S \equiv S'$$

$S'$

$T_1$	$T_2$
-------	-------

$R(A)$   
 $W(A)$

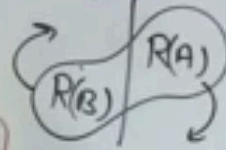
$W(A)$

$R(A)$

$R(A)$

$W(A)$

$W(A)$



$T_1$	$T_2$
-------	-------

$R(A)$

$W(A)$

$W(A)$

$R(B)$

$R(A)$

$T_1$	$T_2$
-------	-------

$R(A)$

$W(A)$

$R(B)$

$R(A)$

$W(A)$

Adjacent Non Conflict pairs



"Conflict Equivalent"

$$S \equiv S'$$

$S'$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
	$R(A)$
	$R(B)$
	$W(A)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
	$R(A)$
$W(A)$	
$R(B)$	

$T_1$	$T_2$
$R(A)$	
$W(A)$	
$R(B)$	
	$R(A)$
	$W(A)$

Adjacent Non Conflict pairs



"Conflict Equivalent"  $S \xrightarrow{CE} S' \rightarrow \text{Serializable}$   
 $S \equiv S'$

$R(A)$   $R(A)$   
 $R(A)$   $W(A)$   
 $W(A)$   $R(A)$   
 $W(A)$

$T_1$	$T_2$
$R(A)$	
$W(A)$	
	$R(A)$
$W(A)$	
$R(B)$	

Change in  
No Positions

$T_1$	$T_2$
$R(A)$	
$W(A)$	
	$R(B)$



Lec-84: Conflict Serializability |  
 Precedence Graph | Transaction | DB...

## Conflict Serializability

800K+ Views

With Example

# Conflict Serializability

## Conflict Serializability

- Check Conflict pairs in other transactions and draw edge

$T_1$	$T_2$	$T_3$
$R(x)$		$R(y)$
		$R(x)$
	$R(y)$	
	$R(z)$	
	$w(z)$	$w(y)$
$R(z)$		
$w(x)$		
$w(z)$		

# Conflict Serializability

R-W  
W-R  
W-W

w(x)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
R(x)		R(y) R(x)
	R(y) R(z)	w(y)
w(x) w(z)		

- Check Conflict pairs in other transactions and draw edges
- Precedence graph

T<sub>1</sub>

T<sub>2</sub>

T<sub>3</sub>



# Conflict Serializability

R-W  
W-R  
W-W  
W(x)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
R(x)		R(y) R(x)
	R(y) R(z)	W(y)
	W(x)	
R(z) W(x) W(z)		

Check Conflict pairs in other transactions and draw edges  
Precedence graph

T<sub>2</sub>



# Conflict Serializability

R-W  
W-R  
W-W

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
		R(y)
		R(x)
	R(y)	
	R(z)	
	W(z)	W(y)
R(z)		
W(x)		
W(z)		

- Check Conflict pairs in other transactions and draw edges
- Precedence graph

(T<sub>1</sub>)

(T<sub>2</sub>)

(T<sub>3</sub>)

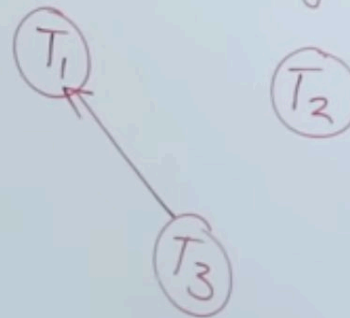
# Conflict Serializability

R-W  
W-R  
W-W

w(x)

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
R(x)		R(y) <del>R(x)</del>
	R(y) R(z)	w(y)
	w(z)	
R(z) W(x) W(z)		

- Check Conflict pairs in other transactions and draw edges  
Precedence graph

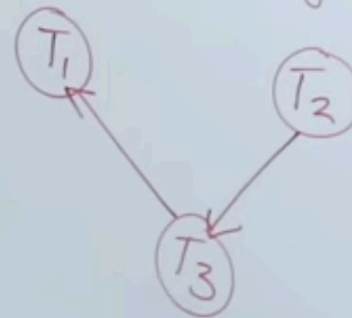


# Conflict Serializability

R-W  
 W-R  
 W-W  
 (W(x))

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
R(x)		R(y) <del>R(x)</del>
	<del>R(y)</del> R(z)	w(y)
	w(z)	
R(z) W(x) W(z)		

- Check Conflict pairs in other transactions and draw edges  
Precedence graph

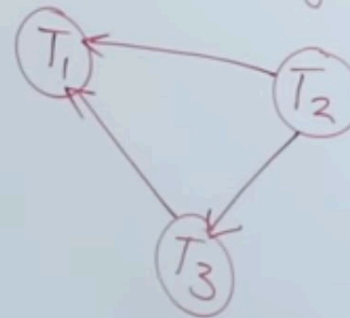


# Conflict Serializability

R-W  
W-R  
W-W  
*(xm)*

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
<del>R(x)</del>		R(y) <del>R(x)</del>
	<del>R(y)</del> <del>R(z)</del>	w(y)
	w(z)	
R(z) W(x) W(z)		

- Check Conflict pairs in other transactions and draw edges  
Precedence graph



# Conflict Serializability

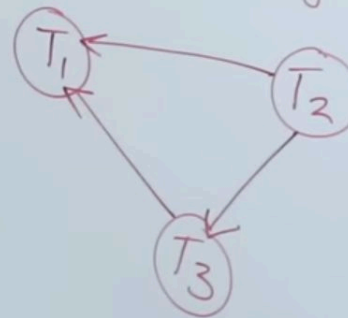
R-W  
W-R  
W-W

$w(x)$

$T_1$	$T_2$	$T_3$
<del><math>R(x)</math></del>		$R(y)$ <del><math>R(x)</math></del>
	<del><math>R(z)</math></del>	
	$w(z)$	
$R(z)$ $w(x)$ $w(z)$		

$w(y) < \begin{matrix} R(y) \\ w(y) \end{matrix}$

- Check Conflict pairs in other transactions and draw edges  
Precedence graph

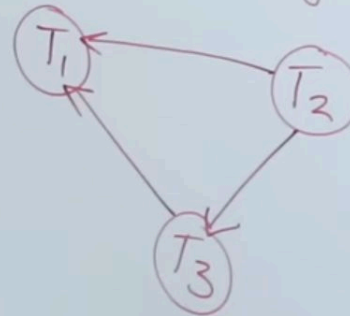


# Conflict Serializability

$W$   
 $R$   
 $W$   
 $W$

$T_1$	$T_2$	$T_3$
<del><math>R(x)</math></del>		$R(y)$ <del><math>R(x)</math></del>
	<del><math>R(y)</math></del> <del><math>R(z)</math></del>	
	$W(z)$	<del><math>W(y)</math></del> $R(y)$ $W(y)$
$R(z)$ $W(x)$ $W(z)$		

- Check Conflict pairs in other transactions and draw edges  
Precedence graph

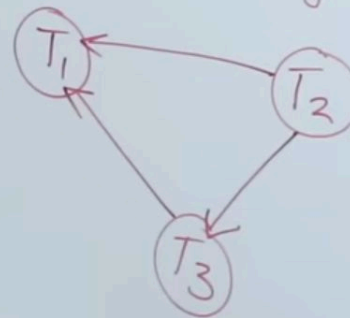


# Conflict Serializability

R-W  
W-R  
W-W  
*w(x)*

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
R(x)		R(y) R(x)
	R(y) R(z)	
	<del>W(z)</del>	
R(z) W(x) W(z)		

- Check Conflict pairs in other transactions and draw edges  
Precedence graph



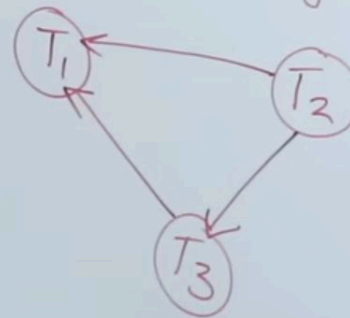


# Conflict Serializability

$R-W$   
 $R-R$   
 $W-W$   
 $w(x)$

$T_1$	$T_2$	$T_3$
<del><math>R(x)</math></del>		$R(y)$ <del><math>R(x)</math></del>
	<del><math>R(y)</math></del> <del><math>R(z)</math></del>	
	<del><math>W(z)</math></del>	
$R(z)$ $W(x)$ $W(z)$		

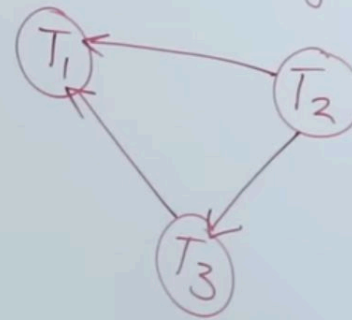
- Check Conflict pairs in other transactions and draw edges  
Precedence graph



## Conflict Serializability

$T_1$	$T_2$	$T_3$
<del><math>R(x)</math></del>		
	$R(y)$ <del><math>R(x)</math></del>	
		$R(y)$ <del><math>w(y)</math></del>

- Check Conflict pairs in other transactions and draw edges
- Precedence graph

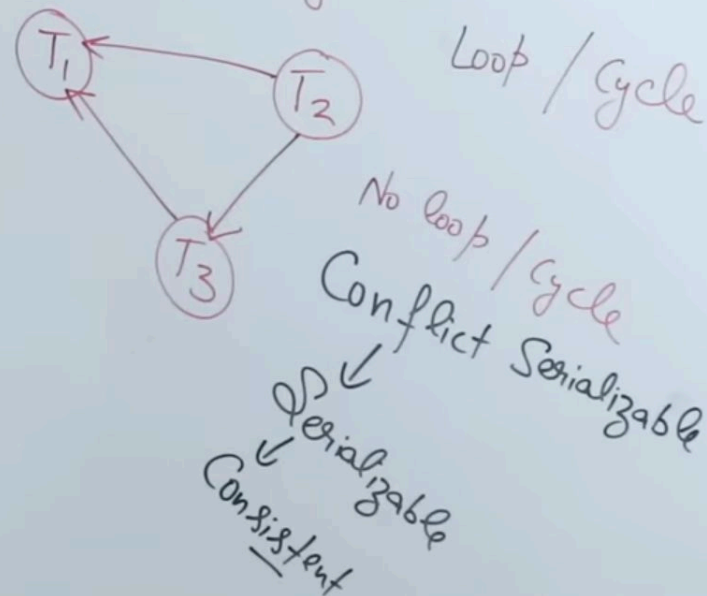


Loop / Cycle

# Conflict Serializability

$T_1$	$T_2$	$T_3$
$R(x)$		$R(y)$ $R(x)$
	$R(y)$ $R(z)$	
	$w(z)$	$w(y) < R(y)$ $w(y)$

- Check Conflict pairs in other transactions and draw edges
- Precedence graph



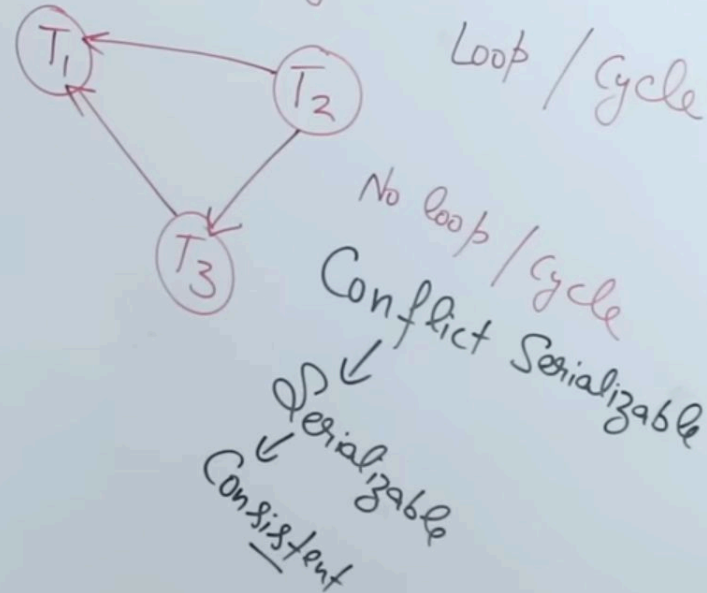
# Conflict Serializability

R-W  
W-R  
W-W

$T_1 \rightarrow T_2 \rightarrow T_3$   
 $T_1 \rightarrow T_3 \rightarrow T_2$   
 $T_2 \rightarrow T_1 \rightarrow T_3$   
 $T_2 \rightarrow T_3 \rightarrow T_1$   
 $T_3 \rightarrow T_1 \rightarrow T_2$   
 $T_3 \rightarrow T_2 \rightarrow T_1$

$T_1$	$T_2$	$T_3$
$R(x)$		$R(y)$ $R(x)$
	$R(y)$ $R(z)$	
	$W(z)$	$W(y) < R(y)$ $W(y)$
$R(z)$ $W(x)$ $W(z)$		

- Check Conflict pairs in other transactions and draw edges  
 Precedence graph

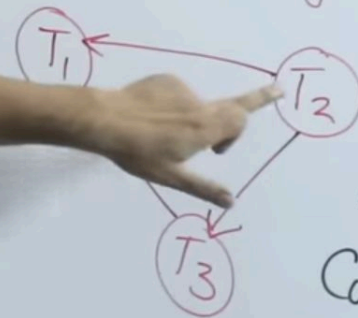


# Conflict Serializability

Indegree = 0

- Check Conflict pairs in other transactions and draw edges

Precedence graph



Loop / Cycle

No loop / cycle

Conflict Serializable

↓  
Serializable  
↓  
Consistent

$w(x) < R(y)$   
 $w(y)$

	T <sub>2</sub>	T <sub>3</sub>
R(x)		
R(y)		R(y)
R(z)		
w(z)		



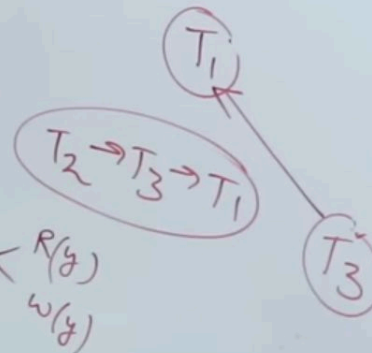
# Conflict Serializability $\text{Indegree} = 0$

- Check Conflict pairs in other transactions and draw edges  
Precedence graph

R-W  
W-R  
W-W

$T_1 \rightarrow T_2 \rightarrow T_3$   
 $T_1 \rightarrow T_3 \rightarrow T_2$   
 $T_2 \rightarrow T_1 \rightarrow T_3$   
 $T_2 \rightarrow T_3 \rightarrow T_1$   
 $T_3 \rightarrow T_1 \rightarrow T_2$   
 $T_3 \rightarrow T_2 \rightarrow T_1$

$T_1$	$T_2$	$T_3$
$R(x)$		$R(y)$ $R(x)$
	$R(y)$ $R(z)$	$w(y) < R(y)$ $w(y)$
	$w(z)$	
		$R(z)$ $w(x)$ $w(z)$



Loop / Cycle

No loop / cycle  
Conflict Serializable

Serializable  
Consistent



# Lec-84: Conflict Serializability | Precedence Graph | Transaction | D



Gate Smashers

Conflict Serializability  $\text{Integrity} = 0$

- Check Conflict pairs in other transactions and draw edges  
Precedence graph

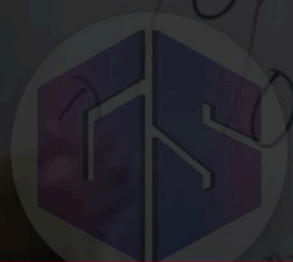
R-W  
W-R  
W-W

$T_1$	$T_2$	$T_3$
$R(x)$		

$T_1/T_2/T_3$



Loop / Cycle



12:08 / 12:17

Lec-85: Why View Serializability is Used | Introduction to View Serializ...

## View Serializability

600K+ Views

Why it is Used?



More videos

Tap or swipe up to see all





# View Serializable

Check whether Schedule is Conflict Serializable  
Or not ?

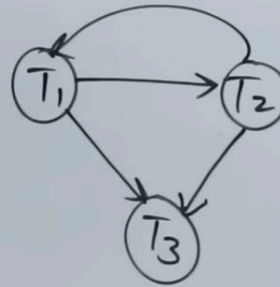
$T_1$	$T_2$	$T_3$
$R(A)$		
	$W(A)$	
$W(A)$		
		$W(A)$

$$W_A < \begin{matrix} R(A) \\ W(A) \end{matrix}$$

S

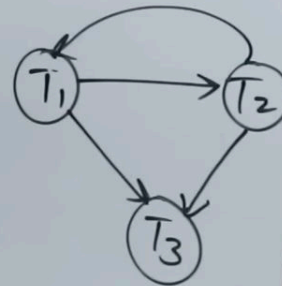
$T_1$	$T_2$	$T_3$
$\times R(A)$		
	$W(A)$	
$W(A)$		
		$W(A)$

Check wheather Schedule is Conflict Serializable  
Or not ?



$WA < R(A)$   
 $WA < W(A)$

Check whether Schedule is Conflict Serializable or not?



Non Conflict Serializable

No Loop / Cycle

↓  
CS

↓  
Serial

↓  
Consistent

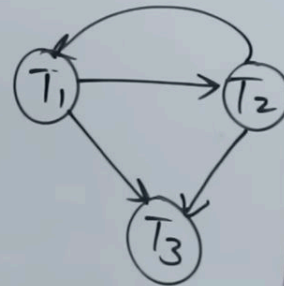
$R(A)$   
 $W(A)$

S

	$T_2$	$T_3$
$R(A)$		
$W(A)$		

View

Check whether Schedule is Conflict Serializable or not?



Non Conflict Serializable  $\rightarrow$  Answer <sup>x</sup>

No Loop / Cycle

$\downarrow$   
CS

$\downarrow$   
Serial

$\downarrow$   
Consistent

$$WA < R(A)$$

$$A = 100$$

S

$T_1$	$T_2$	$T_3$
100 R(A)	$A = A - 40$ $W(A) - 60$	
$A = A - 40$ $W(A)$ $60 - 40$ $= 20$		$A = A - 20$ $W(A)$

View

Check wheather Schedule is Conflict Serializable  
Or not ?

$T_1$	$T_2$	$T_3$
R(A) W(A)	W(A)	W(A)

Not Serializable  $\rightarrow$  Answer <sup>x</sup>

$$WA < R(A)$$

A = 100 S

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
100 R(A)	A = A - 40 W(A) - 60	
A = A - 40 W(A) 60 - 40 = 20		A = A - 20 W(A) 0 ✓

View

Check wheather Schedule is Conflict Serializable  
Or not ?

A = 100

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
100 R(A) 60 W(A)	20 W(A)	0 ✓ W(A)

± Serializable → Answer<sup>x</sup>



$$WA < R(A)$$

= 100

S

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
R(A)	A = A - 40	
W(A)	W(A) - 60	
		A = A - 20
		W(A)
		0 ✓

A = A - 40  
W(A)  
100 - 40  
60

View

A = 100

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
100 R(A)		
60 W(A)	20 W(A)	0 W(A)

Check wheather Schedule is Conflict Serializable Or not ?

Not Serializable → Answer

