

Para el esquema de ejecución concurrente de la figura copiada a continuación se pide dibujar el grafo *read antes que writes* para determinar si la ejecución es serializable. También se pide aplicar el protocolo de ejecución optimista de transacciones para determinar cuales transacciones pueden finalizar con éxito bajo este protocolo de sincronización.

T1	T2	T3	T4
read(A)			
		read(B)	
	read(C)		
		write(A)	
read(B)			
			read(B)
write(C)			
	read(B)		
			read(A)
		write(C)	
			write(B)
	write(A)		

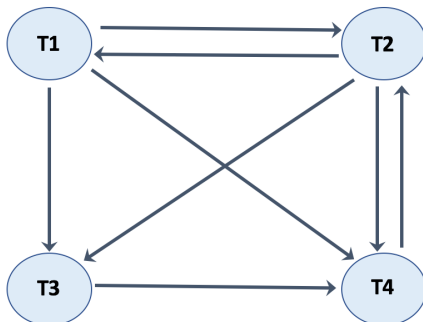
Si T_j lee el valor de un ítem A , que fue escrito por T_i , entonces se crea un arco $T_i \rightarrow T_j$.

Si T_i lee el valor de un ítem A y a continuación T_j lo modifica entonces existe un arco $T_i \rightarrow T_j$.

- $W_{TS}(Q)$: Máximo de los timestamps de las transacciones que han hecho write(Q).
- $R_{TS}(Q)$: Máximo de los timestamps de las transacciones que han hecho read(Q).

El protocolo que asegura la correcta ejecución de las transacciones es el siguiente:

- Si T_i intenta ejecutar read(Q)
 - Si $T_s(i) < W_{TS}(Q)$ no se ejecuta la operación y T_i se aborta.
 - Si $T_s(i) > W_{TS}(Q)$ se ejecuta la operación y $R_{TS}(Q) = \max\{R_{TS}(Q), T_s(i)\}$.
- Si T_i intenta un write(Q)
 - Si $T_s(i) < R_{TS}(Q)$ no se ejecuta la operación y T_i se aborta.
 - Si $T_s(i) < W_{TS}(Q)$ no se ejecuta la operación y T_i continúa.
 - Si no, sí se ejecuta la operación y $W_{TS}(Q) = T_s(i)$.



WTS(A)	0	3		Abort T2
WTS(B)	0	4		
WTS(C)	0	3		Abort T1
RTS(A)	0	1	4	
RTS(B)	0	3	4	
RTS(C)	0	2		