

# Protocolos basados en grafos

- Son una alternativa de protocolos para control de concurrencia basados en bloqueos.
- Imponen un **orden parcial sobre el conjunto D de ítems de datos**
  - $D = \{d_1, d_2, \dots, d_n\}$ , forma un grafo no cíclico.
  - Si en D  $d_i \rightarrow d_j$  entonces, cualquier transacción que necesite acceder a los datos  $d_i, d_j$ . **debe hacerlo respetando ese orden.**

# Protocolo del árbol

## REGLAS

- El único tipo de bloqueo permitido es *lock-X*.
- Dada una transacción  $T_i$ , su primer bloqueo puede ser sobre cualquier dato.
- Después del primer bloqueo,  $T_i$  puede bloquear un dato  $Q$  sólo si  $T_i$  bloquea actualmente al padre de  $Q$ .
- Los datos pueden desbloquearse en cualquier momento.
- Si  $T_i$  bloqueó y desbloqueó un dato  $Q$ , no puede volver a bloquearlo.

# Análisis del protocolo del árbol

## Ventajas

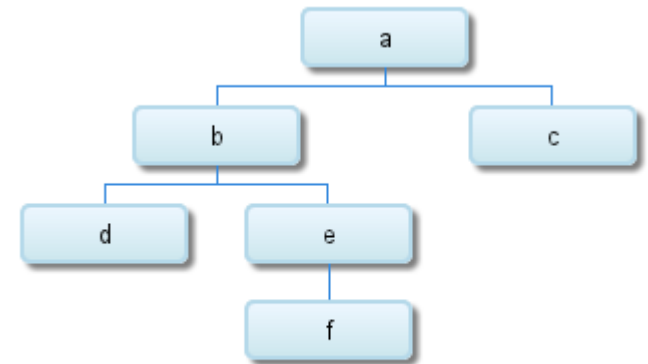
- Asegura serializabilidad en conflictos y está libre de deadlocks.
- Combina secuencias de bloqueos y desbloqueos, *“aumentando la concurrencia”*.

## Desventajas

- No está libre de retrocesos en cascada y planificaciones no recuperables.
- Requiere que se bloqueen datos que no se necesitan, *“disminuye la concurrencia”*.

Tiempo	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
1		Write(D)		
2			Write(E)	
3	Read(C)			
4				Read(D)
5				Write(D)
6		Read(E)		
7			Write(F)	
8	Write(D)			
9		Write(E)		

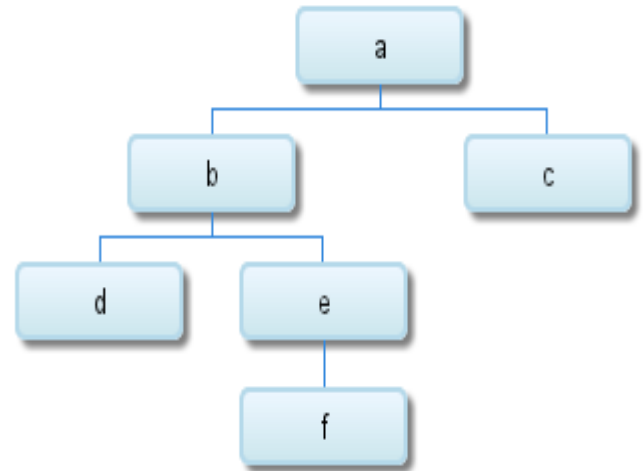
**EJERCICIO 2:** PROTOCOLOS DE GESTION DE CONCURRENCIA (*Arbol*)



¿Es posible obtener esta planificación utilizando protocolo de árbol utilizando las dependencias de datos presentadas? En caso afirmativo, incluya todas las sentencias de bloqueo y desbloqueo necesarias, y en caso negativo indique en qué instante se produce un bloqueo de transacción y cuál es el motivo por el cual se queda bloqueada.

T2 w(d) T3 w(e) T1 r(c) T4 r(d) w(d) T2 r(e) T3 w(f) T1 w(d) T2 w(e)

Tiempo	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
1		Lock_x(b)		
2		Lock_x(d)		
3		WRITE(d)		
4			Lock_x(e)	
5			WRITE(e)	
6	Lock_x(a)			
7	Lock_x(c)			
8	READ(c)			
9		Unlock(d)		
10				Lock_x(d)
11				READ(d)
12				WRITE(d)
13				Unlock(d)



T2 w(d) T3 w(e) T1 r(c) T4 r(d) w(d) T2 r(e) T3 w(f) T1 w(d) T2 w(e)

Tiempo	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
14			Lock_x(f)	
15			Unlock(e)	
16		Lock_x(e)		
17		READ(e)		
18			WRITE(f)	
19		Unlock(b)		
20	Lock_x(b)			
21	Lock_x(d)			
22	WRITE(d)			
23		WRITE(e)		
24		Unlock(e)		
25			Unlock(f)	
26	Unlock(d)			
27	Unlock(a)			

