

## Relación de problemas V: “Gestión y control de la concurrencia”

### Ejercicio 1

Siendo el plan de ejecución inicial el siguiente:

Lee (T3, C), Lee (T1, A), Lee (T2, B), Escribe (T3, B), Lee (T1, B),  
Escribe (T1, A), Escribe (T3, C), Escribe (T2, B)

aplicando el **algoritmo de ordenación parcial modificado** nos encontramos con que:

- al ejecutar Lee(T3,C), ésta se realiza completamente porque no hay ninguna transacción que haya escrito sobre el átomo C,
- al ejecutar lee(T1,A), tampoco ha escrito nadie sobre el átomo A, luego se ejecuta correctamente,
- al ejecutar lee(T2,B), tampoco hay problema porque sobre el átomo B aún no ha escrito nadie,
- al ejecutar Escribe(T3,B), se tiene que el último en leer fue T2, luego se cumple que  $RR(B) = 2 \leq 3$ , y además, como nadie más escribió antes, pues también se cumple que  $WR(B) \leq 3$ , luego se actualiza  $WR(B)$  a 3, y se ejecuta la operación,
- al ejecutar Lee(T1,B), se tiene que justo antes había escrito T3, que es más joven ( $WR(B) \geq 1$ ), luego la **transacción T1 debe abortarse** y llevarse al final del plan (se renombrará a T4),
- al ejecutar Escribe (T3,C), tenemos que  $RR(B) = 2 \leq 3$ , y además que  $WR(B)=3 = 3$ , luego **se ejecuta** la operación y  $WR(B)$  vale 3 (casualmente, el mismo valor, porque la transacción anterior que había escrito también era T3),
- al ejecutar Escribe(T2,B), no aborta porque se tiene que  $RR(B)=2 = 2$ , pero como  $WR(B) = 3 > 2$ , la transacción se ignora y finaliza correctamente;  $WR(B)$  pasa a valer 2.

Ahora ejecutamos la transacción **T4** (es decir, T1 *abortada*), por lo que hay que hacer tres operaciones en este orden: lee, lee, escribe:

- se ejecuta Lee(T4,A), que se realiza correctamente porque nadie ha escrito en A,  $RR(A)$  pasa a valer 4,
- se ejecuta Lee(T4, B), que también se hace correctamente, y  $RR(B)$  pasa a valer 4,
- se ejecuta Escribe(T4, A), y se hace correctamente porque  $RR(A) = 4 = 4$ , y como nadie ha escrito sobre A, se actualiza  $WR(A) = 4$ , se realiza la operación.
- Fin de la secuencia.

En definitiva, el **orden de ejecución de las transacciones es (por el momento en que terminan):**  
**T3, T2, T1.**

## Ejercicio 2

Siendo el plan original:

Lee (T3, C), Lee (T1, A), Lee (T2, B), Escribe (T3, B), Lee (T1, B),  
Escribe (T1, A), Escribe (T3, C), Escribe (T2, B)

Para aplicar el método de bloqueo en dos fases, tenemos que completar el plan con los LOCK y UNLOCK donde corresponda:

LOCK(T3,C,M4), Lee (T3, C), LOCK(T1,A,M4), Lee (T1, A), LOCK(T2,B,M4),  
Lee (T2, B), LOCK(T3,B,M4), Escribe (T3, B), LOCK(T1,B,M1), Lee (T1, B),  
Escribe (T1, A), UNLOCK(T1,A), UNLOCK(T1,B), Escribe (T3, C),  
UNLOCK(T3,C), UNLOCK(T3,B), Escribe (T2, B), UNLOCK(T2,B).

Dicho esto, si aplicamos el algoritmo de bloqueo tenemos que:

- Inicialmente, T3 solicita el bloqueo para escritura de C, y lo logra, por tanto empieza a ejecutarse la lectura. Seguidamente T1 bloqueará A y hará la lectura satisfactoriamente. T2, por su parte, también logrará bloquear el átomo B y hará la lectura.
- **Cuando T3 intenta bloquear el átomo B para escritura (M4) será bloqueado**, porque el modo (M4) es incompatible con el modo de la transacción que ha bloqueado ya a B (M4). Por tanto, T3 se detendrá inmediatamente hasta que T2 haga el unlock sobre el átomo B.
- Mientras ocurre esto, sigue ejecutándose T1, que también quiere acceder al átomo B, sólo que ahora lo quiere hacer sólo para lectura, es decir, en modo M1, el cual es compatible con M4, luego logrará acceso a B y hará su lectura inmediatamente de B; a continuación hará la lectura del átomo A, que ya había bloqueado al principio del plan y como ya no tiene que hacer más operaciones esta transacción liberará los dos átomos que está usando, es decir, ejecutará UNLOCK(T1,B), UNLOCK(T1,A).
- Ahora debería ejecutarse un Escribe(T3,C), pero como T3 está bloqueada hasta que T2 no libere B, será T2 la que actúe, que es la que está después, haciendo su Escribe (T2,B), y justo después, ya que ha acabado hará un UNLOCK(T2,B). este UNLOCK, como ya adelantábamos, ha “despertado” a T3 y le permite ya acceder libremente al átomo B; entonces, T3 bloquea al átomo B (LOCK(T3,B,M4)), hace el Escribe (T3,B), Escribe (T3,C) y liberará B y C ejecutando: UNLOCK(T3,B), UNLOCK(T3,C).
- Fin del plan.

De esta forma, el orden en que se ejecutarían las transacciones sería (según el instante en que acaban): **T1, T2, T3**.

### Ejercicio 3

El plan inicial propuesto era:

Lee (T3, C), Lee (T1, A), Lee (T2, B), Escribe (T3, B), Lee (T1, B),  
Escribe (T1, A), Escribe (T3, C), Escribe (T2, B)

si aplicamos el **algoritmo de ordenación total**, se tiene que:

- se puede ejecutar la lectura Lee(T3,C), porque nadie ha escrito antes,
- también se hace la lectura Lee(T1,A),
- también se hace la lectura Lee(T2,B),
- cuando se intenta realizar el Escribe(T3,B), se tiene que  $R(B)=2 < 3$ , luego la marca de tiempo de B pasa a valer 3 y se ejecuta la escritura,
- cuando se intenta realizar la lectura Lee(T1,B), hay que abortar la T1, porque el sello temporal de B es  $2 > 1$  (T2 es la última que actuó sobre B), luego T1 se ejecutará tras la última transacción del plan,  
el *nuevo plan*, "reestructurado", queda así:  
*Lee (T3, C), Lee (T2, B), Escribe (T3, B), Escribe (T3, C), Escribe (T2, B), Lee (T4, A), Lee (T4, B), Escribe (T4, A)*
- ahora, de acuerdo al plan modificado, hay que hacer el Escribe(T3,C), el cual se hará correctamente porque  $R(C) = 3 = 3$ . Su marca temporal se actualiza:  $R(C) = 3$  (lo mismo);
- a continuación se intenta hacer el Escribe(T2,B), pero como la marca de B es  $3 > 2$ , abortaremos T2, y ésta se ejecutará, al completo y en el orden indicado en el plan original, al final del mismo;  
el nuevo plan queda como sigue:  
*Lee (T3, C), Escribe (T3, B), Escribe (T3, C), Lee (T4, A), Lee (T4, B), Escribe (T4, A), Lee (T5, B), Escribe (T5, B).*
- Ahora tenemos que ejecutar las operaciones de **T4** (T1 abortada):
  - se realiza la primera lectura Lee(T4,A) porque  $R(A) = 1 < 4$ ;
  - a continuación se ejecuta Lee(T4,B) porque  $R(B) = 3 < 4$ , y  $R(B)$  pasa a valer 4;
  - también se ejecuta el Escribe(T4,A) correctamente porque  $R(A) = 4 = 4$ ;
- y finalmente intentamos ejecutar la transacción **T5** (T2 abortada):
  - se ejecutará completamente sin errores porque cualquier marca de tiempo de A,B,C será inferior a 5

Es decir, el orden en el que se ejecutarían las transacciones siguiendo el algoritmo de ordenación total sería: **T3, T1, T2**.

## Ejercicio 4

Siendo el plan:

Lee (T1, A), Lee (T2, A), Escribe (T1, A=20), Lee (T3, B), Escribe (T2, A=30), Escribe (T3, B=15), Escribe (T3, D=25), Escribe (T2, E=35)

Si aplicamos el algoritmo de ordenación parcial, se tiene que:

- al ejecutar Lee(T1,A), se hace correctamente,
- al ejecutar Lee (T2, A) se realiza sin problemas,
- al ejecutar Escribe(T1,A=20), se tiene que  $RR(A) = 2 > 1$ , luego la transacción T1 tendrá que ser abortada, y llevada al final de plan. El nuevo plan queda así:

Lee (T2, A), Lee (T3, B), Escribe (T2, A=30), Escribe (T3, B=15), Escribe (T3, D=25), Escribe (T2, E=35), **Lee (T1, A), Escribe (T1, A=20)**.

- Al ejecutar Lee(T3,B), ésta se hace sin problemas porque nadie ha escrito aún sobre B,
- Al ejecutar Escribe(T2,A=30), ésta se realizará sin problemas también porque  $RR(A)=2 = 2$  y además ninguna transacción ha escrito antes sobre A,
- Al ejecutar Escribe(T3,B=15), pasa lo mismo:  $RR(B)=3 = 3$  y nadie ha escrito aún sobre B,
- Al ejecutar Escribe(T3,D), se ejecuta sin problemas porque nadie ha leído o escrito antes sobre D,
- Al ejecutar Escribe(T2,E), ésta se realiza sin problemas de nuevo porque nadie había leído o escrito antes sobre E.
- Finalmente, se ejecutará la transacción T4 (T1 abortada):
  - Lee(T1,A) se realiza sin problemas porque  $WR(A) < 4$ ,
  - Escribe(T1,A=20), también se realiza correctamente porque  $RR(A) < 4$  Y  $WR(A) < 4$ .

En definitiva, el orden de ejecución de las transacciones (por el instante en que acaban) será:

**T3, T2, T1.**

## Ejercicio 5

Tendremos que reescribir el plan para añadir los LOCK y UNLOCK donde corresponda:

Bloqueo (T1, A, M5), Lee (T1, A), Bloqueo (T2, A, M5), Lee (T2, A),  
Escribe (T1, A=20), Desbloqueo (T1, A), Bloqueo (T3, B, M5), Lee (T3,  
B), Escribe (T2, A=30), Escribe (T3, B=15), Bloqueo (T3, D, M5), Escribe  
(T3, D=25), Desbloqueo (T3, B), Desbloqueo (T3, D), Bloqueo (T2, E, M5),  
Escribe (T2, E=35), Desbloqueo (T2, A), Desbloqueo (T2, E).

Entonces, siguiendo el algoritmo, se tiene que:

1. Al principio, T1 consigue bloquear el átomo A, y puede realizar el Lee(T1,A), y bloquea a T2 cuando ésta intenta bloquear a dicho átomo ya que el modo M5 con el que T1 está accediendo es incompatible con el modo M5 que necesita la t2 sobre dicho átomo.
2. Pero mientras que T2 está paralizada, se sigue ejecutando T1, hasta llegar a la instrucción de desbloqueo del átomo A. Este desbloqueo provoca que T2 se reactive, bloqueando el átomo A en el modo M5 y pudiendo realizar su operación sobre dicho átomo. El plan se ha modificado de la siguiente forma:

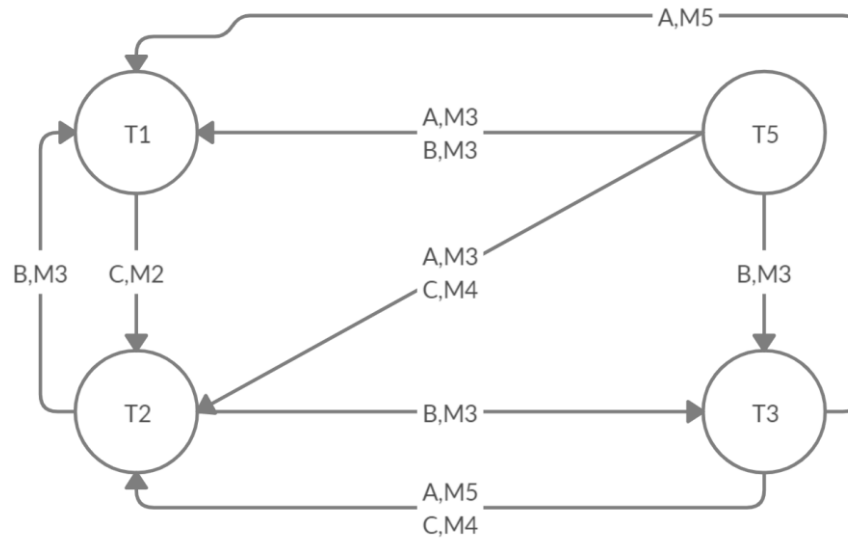
Bloqueo (T1, A, M5), Lee (T1, A), Escribe (T1, A=20), Desbloqueo  
(T1, A), **Bloqueo (T2, A, M5), Lee (T2, A)**, Bloqueo (T3, B, M5),  
Lee (T3, B), Escribe (T2, A=30), Escribe (T3, B=15), Bloqueo (T3,  
D, M5), Escribe (T3, D=25), Desbloqueo (T3, B), Desbloqueo (T3,  
D), Bloqueo (T2, E, M5), Escribe (T2, E=35), Desbloqueo (T2, A),  
Desbloqueo (T2, E)

3. Ahora T3 bloquea el átomo B en modo M5 y consigue bloquear el átomo D en el mismo modo y seguir ejecutándose hasta llegar a su última operación y, como es procedente, desbloquear ambos átomos (UNLOCK de B y D).
4. De la misma manera, la transacción 2 consigue bloquear el átomo E en modo M5 y termina su ejecución desbloqueando los átomos A y E.

En conclusión, el orden de ejecución de las transacciones es **T1, T3, T2**.

## Ejercicio 7

El grafo de bloqueo para la situación planteada es:



En el grafo podemos observar que hay varios ciclos, por tanto se producirán bloqueos mortales en algunos casos.