Ferhat Elmas 214805

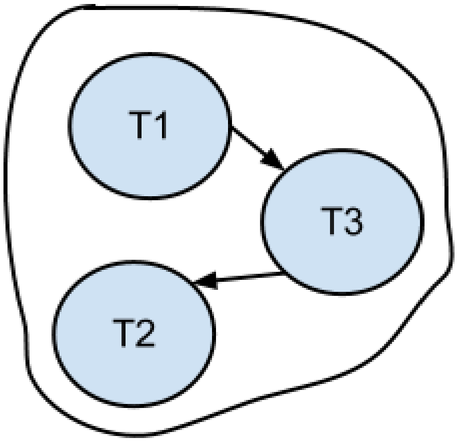
Advanced Databases Hw-12

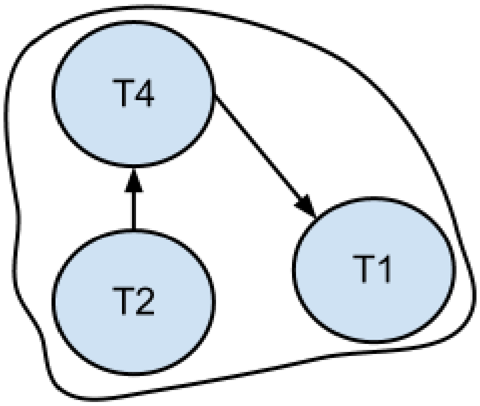
1-) Modified 2PC solves the problem of blocking when the coordinator goes down. In default 2PL, a subordinate that says NO can directly abort but a subordinate that says YES must wait for the response of the coordinator to commit and since coordinator is down, it will be blocked permanently or until coordinator come back. However, if subordinates can talk to each other and there is at least one NO, they don't need to wait for the coordinator to abort. Modified 2PL can't solve blocking problem when all subordinates say YES. Moreover, since all subordinates can talk to each other, this will require and consume more network resources such as bandwidth.

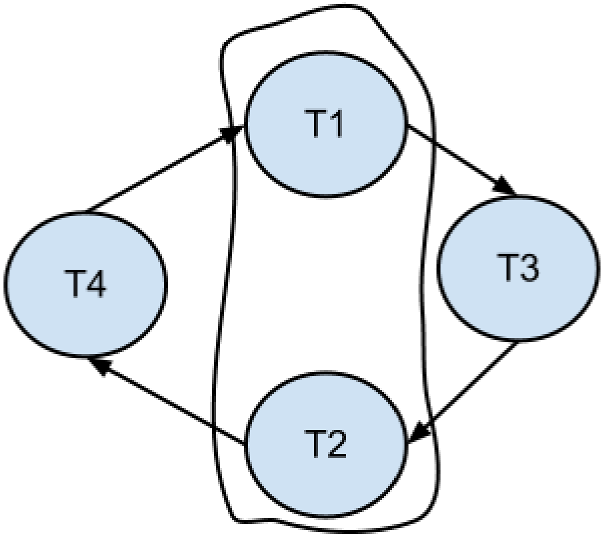
2-)

Schedule at Site S1: r1(A) **c1** w3(A) w3(B) **c3** r2(B) **c2** so T1 -> T3 -> T2 is serializable
Schedule at Site S2: w4(C) r1(C) **c1** r2(D) **c2** w4(d) **c4** so T2 -> T4 -> T1 is serializable.

As you can see below, they aren't globally serializable.







3 -)

1.1 -) Containment mapping from q1 to q1’ is:

Ψ(x3) = y3
Ψ(x5) = y4
Ψ(x4) = y3
Ψ(x1) = y1
Ψ(x2) = y2

However, for mapping from q1’ to q1, Ψ(y3) = x3 and Ψ(x4) but if it returns two values then it can’t be a function.

As a result, $q1 \subseteq q1'$

1.2 -) For mapping from q2 to q2’, x3 must be mapped to y2 but we don’t have any rows that satisfy this mapping. For containment from q2’ to q2 is:

Ψ(y1) = x2
Ψ(y2) = x3

As a result $q2' \subseteq q2$

1.3 -) For mapping from q3 to q3’,

$\Psi(x1) = y2$

$\Psi(x2) = y1$

$\Psi(x3) = y2$

to satisfy the query result but there are some rows that aren’t satisfied with this mapping.

For mapping from q3’ to q3,

$\Psi(y1) = x2$

$\Psi(y2) = x3$ but we don’t have the x3, x3 row.

As a result, neither of the queries contain each other.

4 -)

x	y1	z	w
x1	y1	z1	w1
x	y2	z1	z
x2	y	z	y
x	y	z	

if we have defined a map,

$\Psi(x1) = x$

$\Psi(z1) = z$

$\Psi(w1) = w$

we can get rid of the second row.

This mapping comes from

- First, third and fourth rows cannot be eliminated because every homomorphism is identity.

x	y1	z	w
x	y2	z1	z
x2	y	z	y
x	y	z	

However, we are trying to find a homomorphism onto itself and these mapping can create new rows that aren’t exist in the actual tableau so we should check this mapping result on other rows. For example, third row is mapped to (x, y2, z, z) and this row isn’t in the actual tableau. Therefore, above defined mapping isn’t a valid mapping and as a result of not being able to find a valid mapping, we can’t reduce the table.