

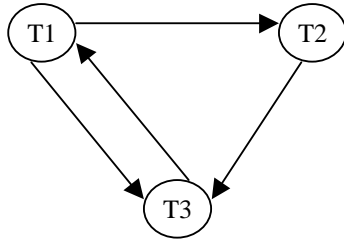
CSC742 Homework6

Solution

19.22

a). $r1(X); r3(X); w1(X); r2(X); w3(X);$

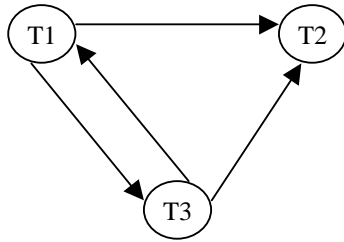
The serialization graph is:



Not serializable.

b). $r1(X); r3(X); w3(X); w1(X); r2(X);$

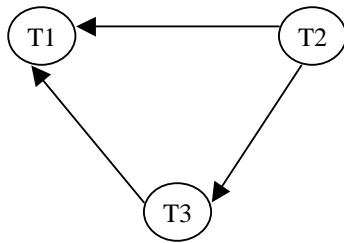
The serialization graph is:



Not serializable.

c). $r3(X); r2(X); w3(X); r1(X); w1(X);$

The serialization graph is:

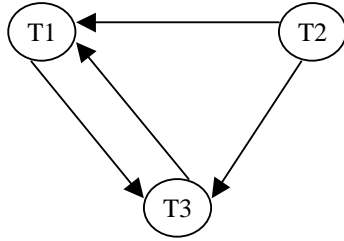


Serializable.

The equivalent serial schedule is: $r2(X); r3(X); w3(X); r1(X); w1(X);$

d). $r3(X); r2(X); r1(X); w3(X); w1(X);$

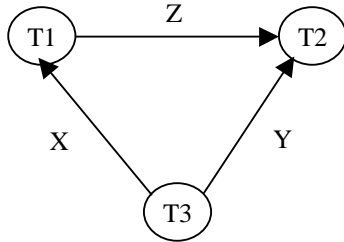
The serialization graph is:



Not serializable.

19.23

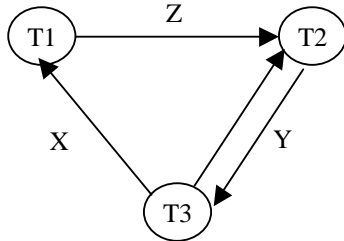
1). The serialization graph for S1 is:



S1 is serializable. The equivalent serial schedule is: (T3, T1, T2)

$r3(X); r3(Y); w3(Y); r1(X); r1(Z); w1(X); r2(Z); r2(Y); w2(Z); w2(Y);$

2). The serialization graph for S2 is:



S2 is not serializable.

19.24

1). $S3: r1(X); r2(Z); r1(Z); r3(X); r3(Y); w1(X); c1; w3(Y); c3; r2(Y); w2(Z); w2(Y); c2;$

In S3, every transaction commits right after it writes some items. There is no write to or read from an item before the last transaction that wrote that item has committed. So S3 is **strict**.

2). $S4: r1(X); r2(Z); r1(Z); r3(X); r3(Y); w1(X); w3(Y); r2(Y); w2(Z); w2(Y); c1; c2; c3;$

In S4, T2 reads item Y from T3 but T2 commits before T3 commits. So S4 is **nonrecoverable**.

3). $S5: r1(X); r2(Z); r3(X); r1(Z); r2(Y); r3(Y); w1(X); c1; w2(Z); w3(Y); w2(Y); c3; c2;$

S5 is **not strict** because T2 writes Y before T3 commits. But S5 is **cascadeless** because there is no transaction reads items that were written by an uncommitted transaction. S5 is not serializable.