

Deadline-6

Hitesh Kumar 2021257 || Sahil Goswami 2021281

Example of 4 Transaction which are Non-Conflict

T1	T2	T3	T4
R(A) A=A-500 W(A)	R(A) A=A-200 W(A)	R(B) R(A) B=B-500 A=A-100 W(B) W(A)	R(A)

This is an example of non-conflict transactions.

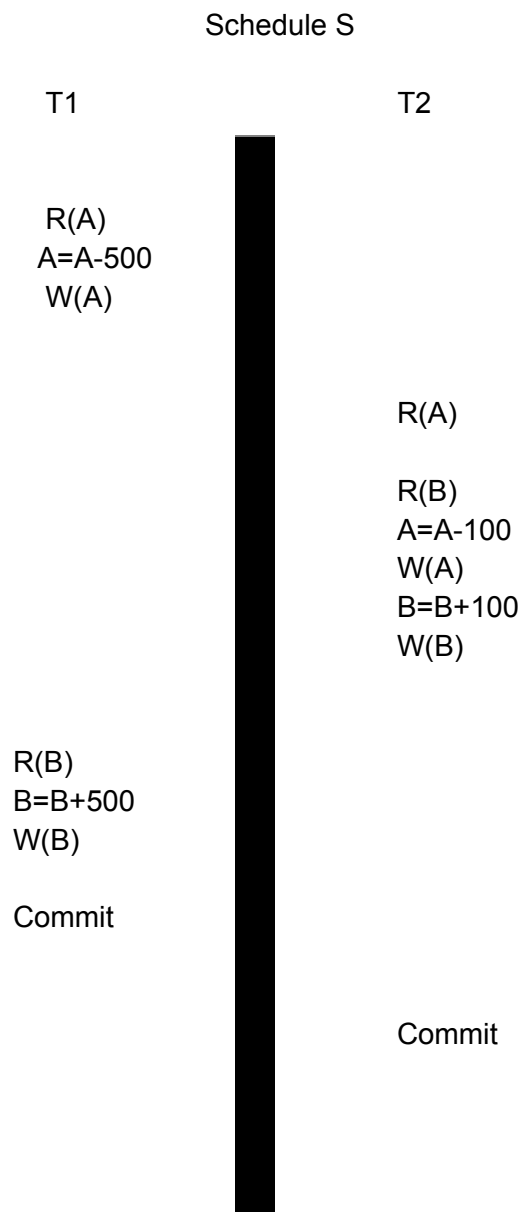
This is not achieved Consistency.

Example of 2 Transaction which are conflict transactions.

T1	T2
R(A) A=A-500 W(A)	R(A) R(B) B=B-500 W(B)

This is conflict Transaction. Here consistency is achieved.

Let us Discuss about a Schedule of 2 Transaction:



This Schedule is Example of non-conflict serializable .

Make precedence graph



Loop is present because R(A) in Transaction T1 is conflicting pair with W(A) in Transaction T2

And W(B) in Transaction T2 is conflicting pair with W(B) in Transaction T1.

So They create a problem and this is an example of non-conflict serializable.

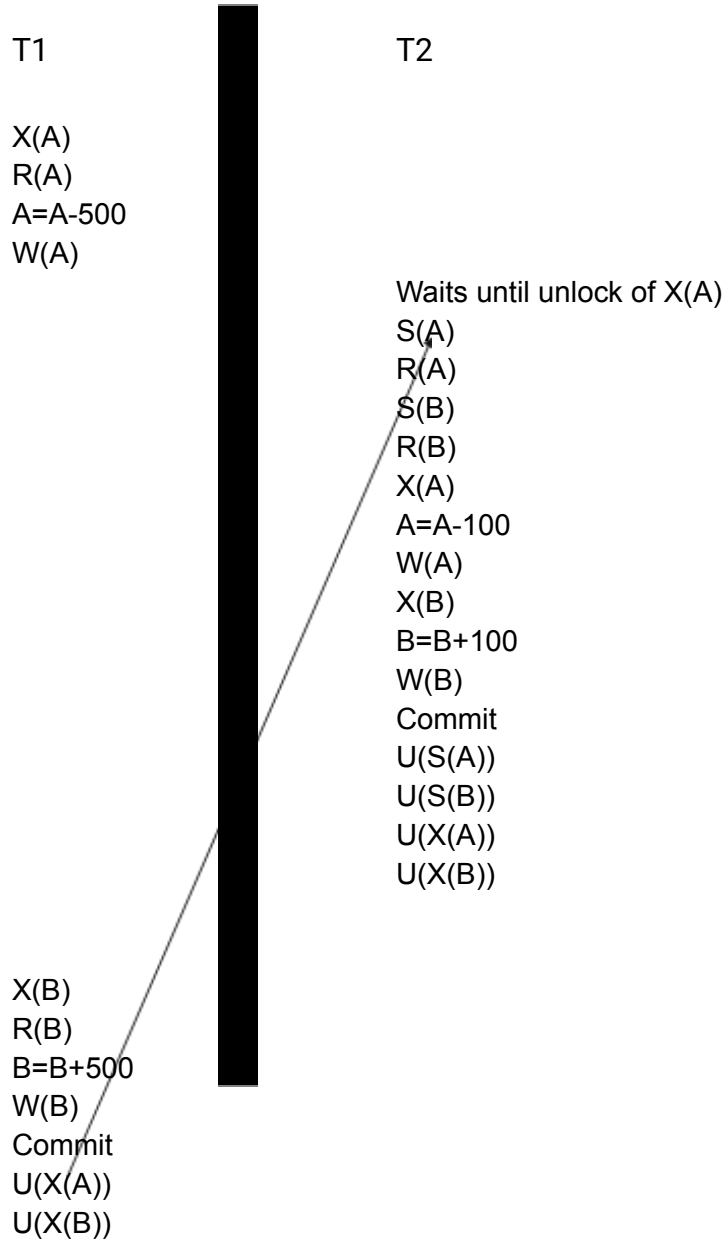
We know that our main motive is to make concurrent transactions in series so that consistency is maintained.

Now how we make the above example to conflict serializable.

We use the concept of locking to do this work.

Here we use Rigorous 2PL locking to make this conflict serializable.

Schedule S1



Where X(A) means exclusive lock in A.

S(A) means shared lock in A.

X(B) means exclusive lock in B.

S(B) means shared lock in B.

U(X(A)) means unlock the exclusive lock in A.

U(X(B)) means unlock the exclusive lock in B.

U(S(A)) means unlock the shared lock in A.

U(S(B)) means unlock the shared lock in B.

This is Now become Conflict Serializable.

1st T1 completes then T2 completes. Now series Schedule is Achieved. Now Consistency is achieved.