Conflict Serializability

T1	T2
read(A)	
A:= A – 50	
write(A)	
	read(A)
	temp := A*0.1
	A:= A – temp
	write(A)
read(B)	
B:= B + 50	
write(B)	
	read(B)
	B:= B + temp
	write(B)

Fig: Schedule-3 (consistent schedule)

→

T1	T2
read(A)	
write(A)	
	read(A)
	write(A)
read(B)	
write(B)	
	read(B)
	write(B)

T1	T2
read(A)	
A:= A – 50	
	read(A)
	temp := A*0.1
	A:= A – temp
	write(A)
	read(B)
write(A)	
read(B)	
B:= B + 50	
write(B)	
	B:= B + temp
	write(B)

Fig: Schedule-4 (inconsistent schedule)

→

T1	T2
read(A)	
	read(A)
	write(A)
	read(B)
write(A)	
read(B)	
write(B)	
	write(B)

Conflict Equivalent and Serializable

T1	T2
read(A)	
write(A)	
	read(A)
	write(A)
read(B)	
write(B)	
	read(B)
	write(B)

Schedule-3

Swap non-conflicting instructions read(B) of T1 with write(A) of T2

T1	T2
read(A)	
write(A)	
	read(A)
read(B)	
	write(A)
write(B)	
	read(B)
	write(B)

Swap non-conflicting instructions read(B) of T1 with read(A) of T2

T1	T2
read(A)	
write(A)	
read(B)	
	read(A)
	write(A)
write(B)	
	read(B)
	write(B)

Swap non-conflicting instructions write(B) of T1 with write(A) of T2

T1	T2
read(A)	
write(A)	
read(B)	
	read(A)
write(B)	
	write(A)
	read(B)
	write(B)

Swap non-conflicting instructions write(B) of T1 with read(A) of T2

T1	T2
read(A)	
write(A)	
read(B)	
write(B)	
	read(A)
	write(A)
	read(B)
	write(B)

Schedule-6 (Serial schedule)

Here, Schedule-3 is Conflict Serializable

Schedule-3 and Schedule-6 is Conflict Equivalent

Summary:

- **Conflict Serializable:**
  - A schedule is called conflict serializable if it can be transformed into a serial schedule by swapping non-conflicting operations.
  - We can say that a **schedule *S* is conflict serializable if it is conflict equivalent to a serial schedule**

- **Conflicting operations:** Two operations are said to be conflicting if all conditions satisfy:
  - They belong to different transactions
  - They operate on the same data item
  - At Least one of them is a write operation

- **Conflict Equivalent*:*** If a schedule *S* can be transformed into a schedule *S´* by a series of **swaps of non-conflicting instructions**, we say that *S* and *S´* are conflict equivalent.

Serial Schedule VS [✓ Serializable Schedule]

Which one is better??

Some more examples:

Example-1

T3	T4
read(Q)	
	write (Q)
write (Q)	

Fig: Schedule-7

Is Schedule-7 Conflict Serializable?????

T3	T4
read(Q)	
	write (Q)
write (Q)	

Fig: Schedule-7 **(not conflict serializable)**

Example-2

**It is possible to have two schedules that produce the same outcome, but that are not conflict equivalent.

T1	T5
read(A)	
A:= A – 50	
write(A)	
	read(B)
	B:= B – 10
	write(B)
read(B)	
B:= B + 50	
write(B)	
	read(A)
	A:= A + 10
	write(A)

Are these two Schedules Consistant??????

Are these two Schedules Conflict Equivalent??????

T1	T5
read(A)	
A:= A – 50	
write(A)	
read(B)	
B:= B + 50	
write(B)	
	read(B)
	B:= B – 10
	write(B)
	read(A)
	A:= A + 10
	write(A)

T1	T5
read(A)	
A:= A – 50	
write(A)	
	read(B)
	B:= B – 10
	write(B)
read(B)	
B:= B + 50	
write(B)	
	read(A)
	A:= A + 10
	write(A)

Fig: Schedule-8

These two schedules produce same outcome, but not Conflict Equivalent

Schedule-8 is not Conflict Serializable.

T1	T5
read(A)	
A:= A – 50	
write(A)	
read(B)	
B:= B + 50	
write(B)	
	read(B)
	B:= B – 10
	write(B)
	read(A)
	A:= A + 10
	write(A)