UNIT IV

TRANSACTIONS

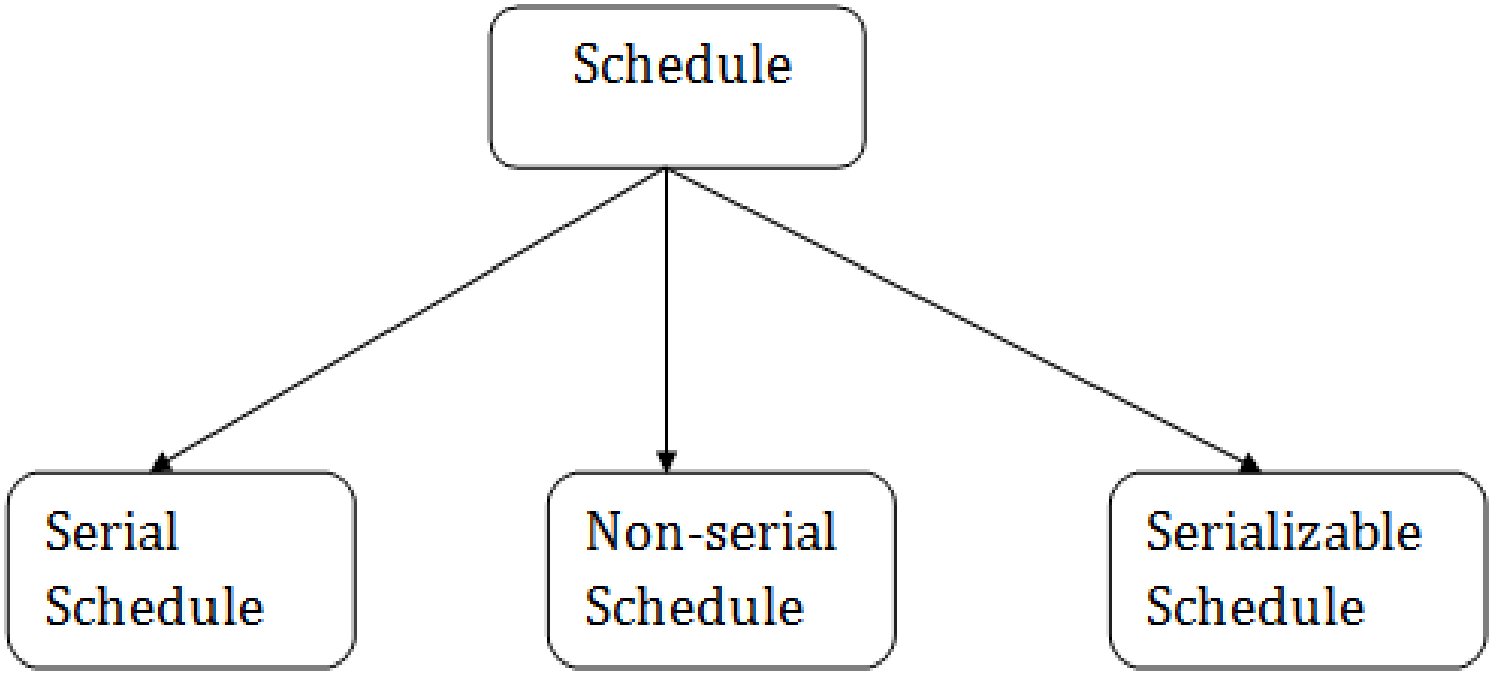
Prepared by,
K.Revathi AP/IT, SNSCE

Schedules

Definition:

A series of operation from one transaction to another transaction is known as schedule. It is used to preserve the order of the operation in each of the individual transaction.

Active state



Serial Schedule

- The serial schedule is a type of schedule where one transaction is executed completely before starting another transaction
- (a) Execute all the operations of T1 which was followed by all the operations of T2.
- (b) Execute all the operations of T2 which was followed by all the operations of T1.

Example

(a)

Time

T1	T2

read(A);
A := A – N;
write(A);
read(B);
B := B + N;
write(B);
	read(A);
	A := A + M;
	write(A);

Schedule A

(b)

Time

T1	T2

	read(A);
	A := A + M;
	write(A);

read(A);
A:= A - N;
write(A);
read(B);
B:= B +N;
write(B);	

Schedule B

2. Non-serial Schedule

- If interleaving of operations is allowed, then there will be non-serial schedule.
- It contains many possible orders in which the system can execute the individual operations of the transactions.

Example

(c)

Time

T1	T2
read(A);	
A := A – N;	
	read(A);
	A := A + M;
write(A);	
read(B);	
	write(A);
B := B + N;	
write(B);	

Schedule C

(d)

T1	T2

read(A);
A := A – N;
write(A);	

	read(A);
	A := A + M;
	write(A);

read(B);
B := B + N;
write(B);	

Time

Schedule D

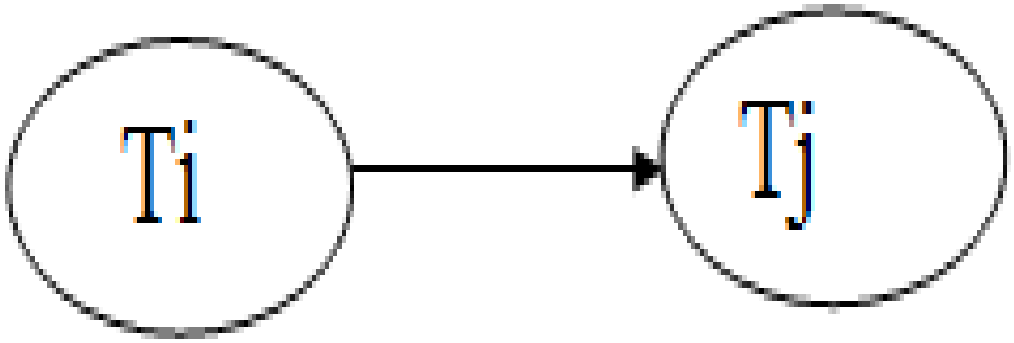
3. Serializable Schedule

- The serializability of schedules is used to find non-serial schedules that allow the transaction to execute concurrently without interfering with one another.
- It identifies which schedules are correct when executions of the transaction have interleaving of their operations.

Testing of Serializability

- Serialization Graph is used to test the Serializability of a schedule.
- Three conditions holds: Ti → Tj
  - Create a node Ti → Tj if Ti executes write (Q) before Tj executes read (Q).
  - Create a node Ti → Tj if Ti executes read (Q) before Tj executes write (Q).
  - Create a node Ti → Tj if Ti executes write (Q) before Tj executes write (Q).

Precedence graph for Schedule S



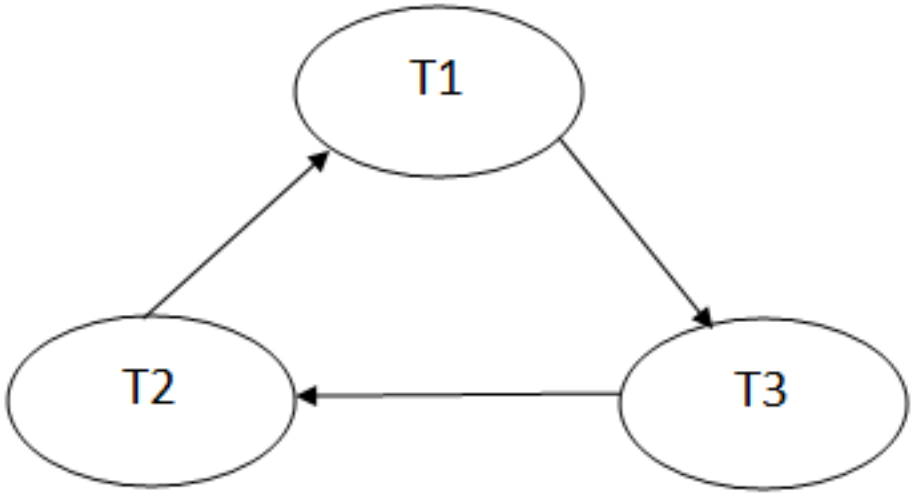
Example

Time

T1	T2	T3
Read(A)		
	Read(B)	
A:= f_1(A)		
		Read(C)
	B:= f_2(B)	
	Write(B)	
		C:= f_3(C)
		Write(C)
Write(A)		
		Read(B)
	Read(A)	
	A:= f_4(A)	
Read(C)		
	Write(A)	
C:= f_5(C)		
Write(C)		
		B:= f_6(B)
		Write(B)

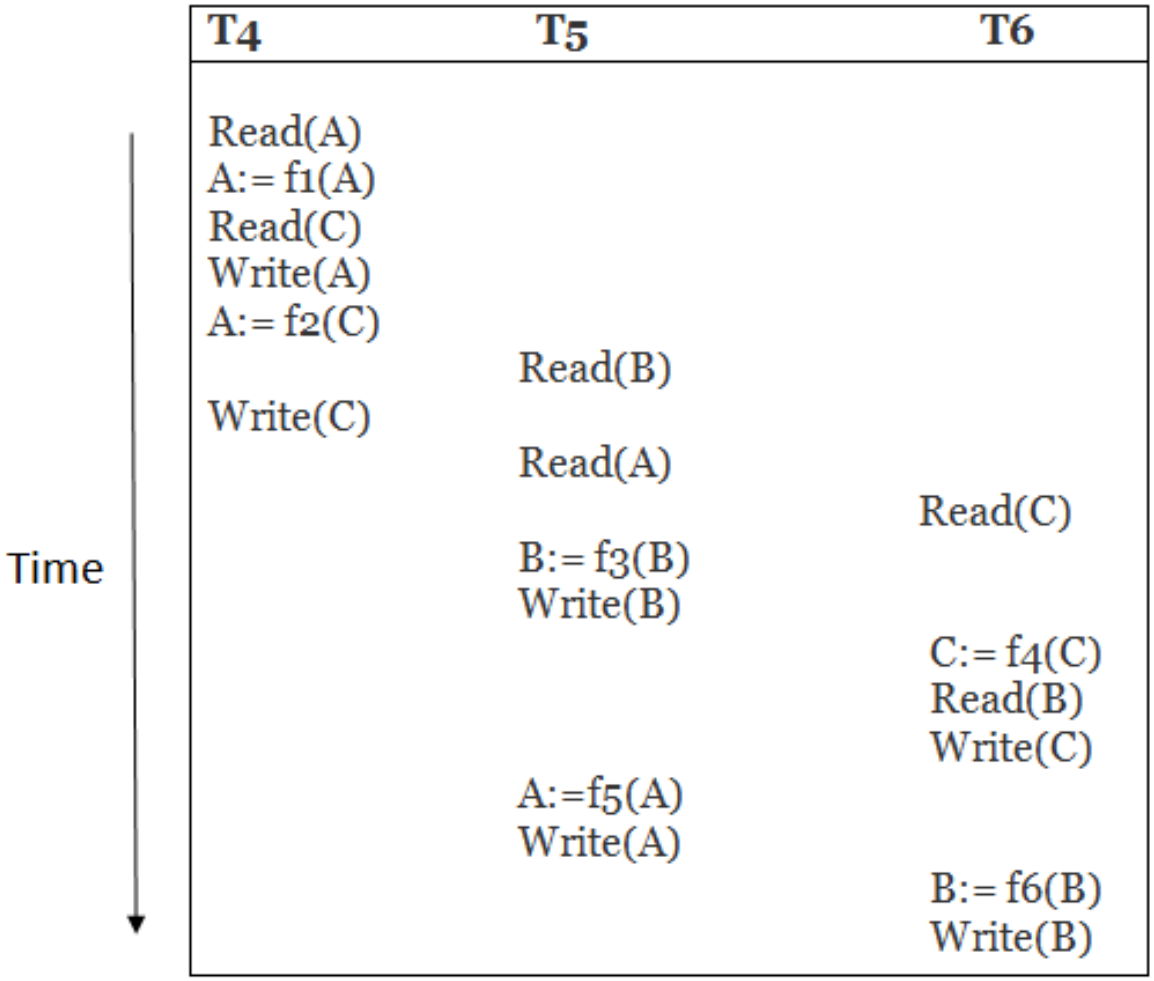
Schedule S1

Precedence graph for schedule S1:



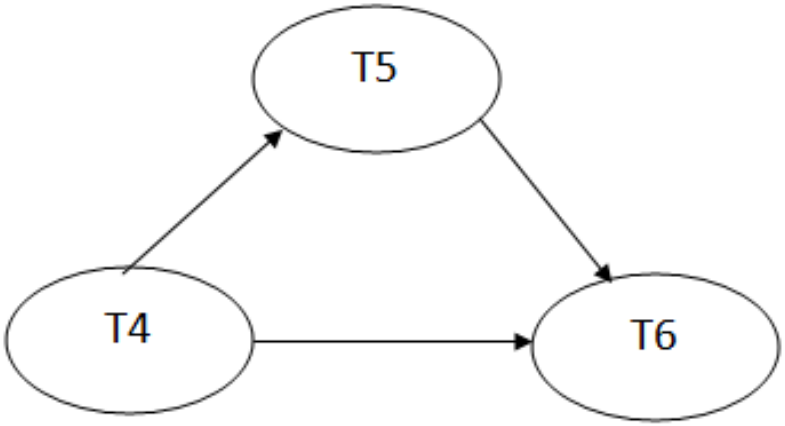
- The precedence graph for schedule S1 contains a cycle
- Schedule S1 is non-serializable.

Example 2



Schedule S2

Precedence graph for schedule S2:



- The precedence graph for schedule S2 contains no cycle
- ScheduleS2 is serializable.