CHAPTER 18: CONCURRENCY CONTROL TECHNIQUES

✓ 18.20 Prove that the basic two-phase locking protocol guarantees conflict serializability of schedules. (Hint: Show that, if a serializability graph for a schedule has a cycle, then at least one of the transactions participating in the schedule does not obey the two-phase locking protocol.)

✓ 18.22 Prove that strict two-phase locking guarantees strict schedules.

✓ 18.24 Prove that cautious waiting avoids deadlock.

18.25 Apply the timestamp ordering algorithm to the schedules of Figure 17.8(b) and (c), and determine whether the algorithm will allow the execution of the schedules.

18.26 Repeat Exercise 18.25, but use the multiversion timestamp ordering method.

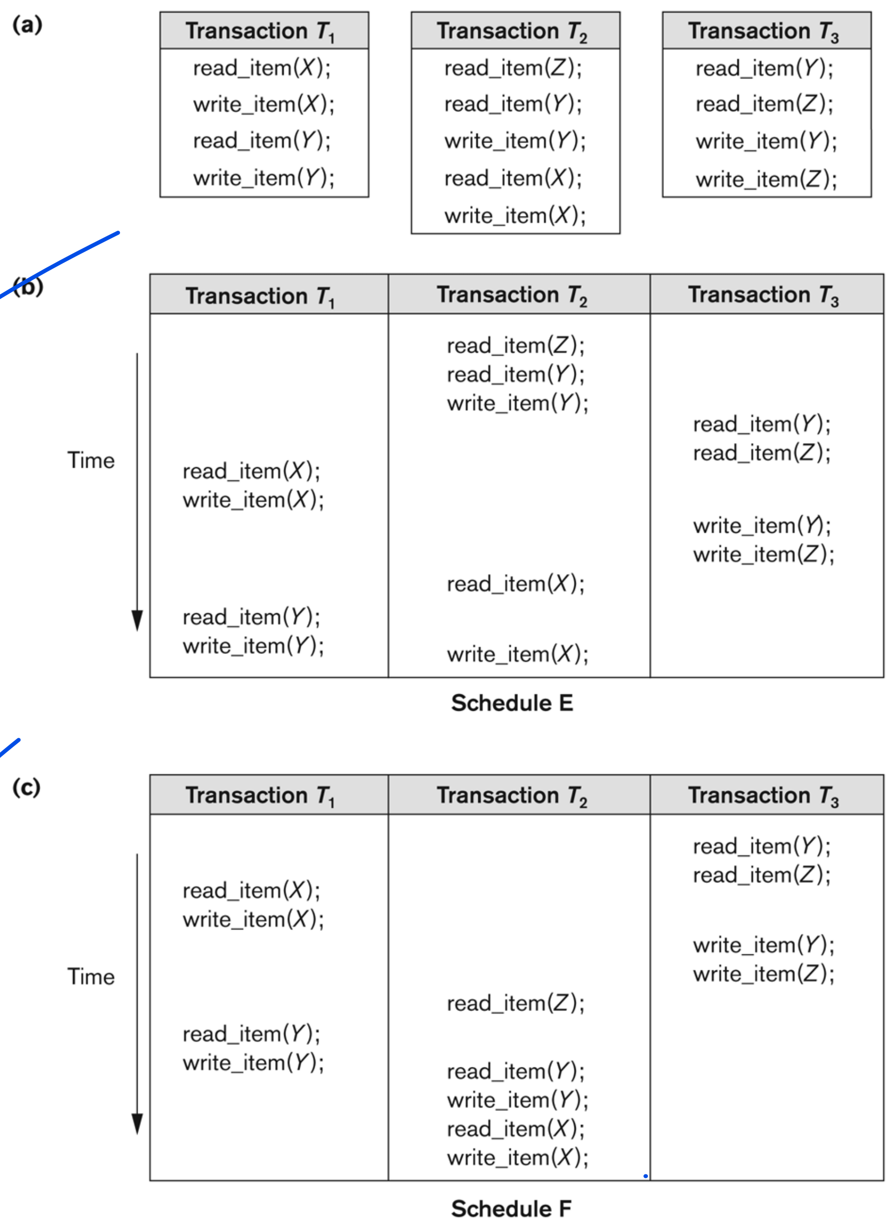

Figure 17.8 Another example of serializability testing. (a) The read and write operations of three transactions T_1, T_2, and T_3. (b) Schedule E. (c) Schedule F.

(a)

Transaction T_1	Transaction T_2	Transaction T_3
read_item(X);	read_item(Z);	read_item(Y);
write_item(X);	read_item(Y);	read_item(Z);
read_item(Y);	write_item(Y);	write_item(Y);
write_item(Y);	read_item(X);	write_item(Z);
	write_item(X);	

✓ (b)

Transaction T_1	Transaction T_2	Transaction T_3
	read_item(Z);	
	read_item(Y);	
	write_item(Y);	
		read_item(Y);
		read_item(Z);
read_item(X);		
write_item(X);		
		write_item(Y);
		write_item(Z);
	read_item(X);	
read_item(Y);		
write_item(Y);		
	write_item(X);	

Time ↓

Schedule E

✓ (c)

Transaction T_1	Transaction T_2	Transaction T_3
		read_item(Y);
		read_item(Z);
read_item(X);		
write_item(X);		
		write_item(Y);
		write_item(Z);
	read_item(Z);	
read_item(Y);		
write_item(Y);		
	read_item(Y);	
	write_item(Y);	
	read_item(X);	
	write_item(X);	

Time ↓

Schedule F

8/1/24 [illegible] [illegible]