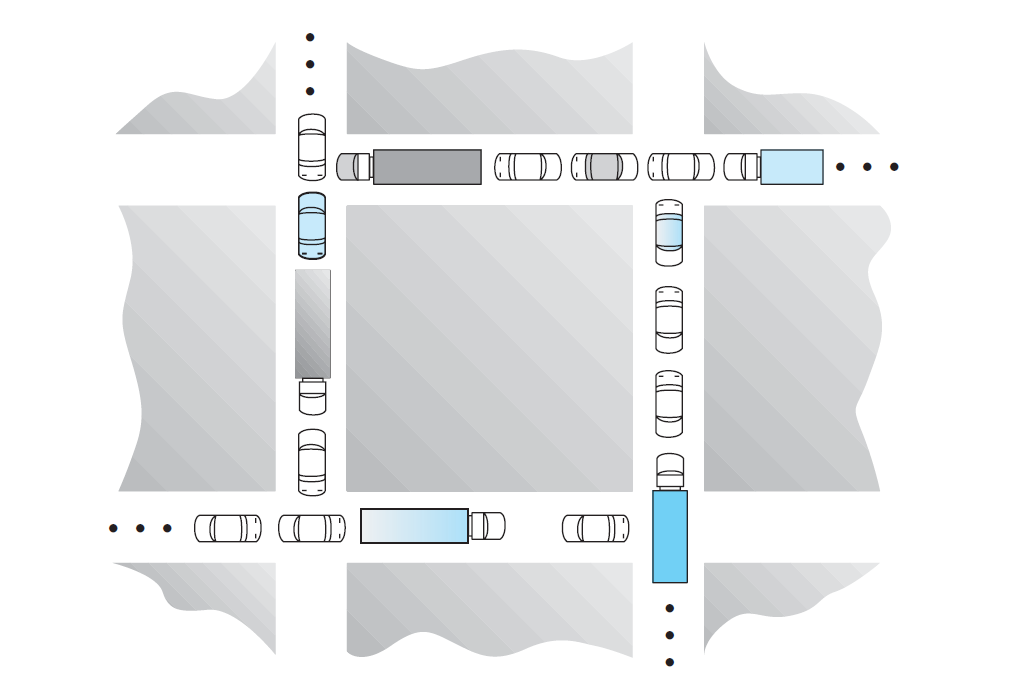
**7.11** Consider the traffic deadlock depicted in Figure 7.10.

a. Show that the four necessary conditions for deadlock hold in this

example.

b. State a simple rule for avoiding deadlocks in this system.



**a.**

**Mutual exclusion**. only one process at a time can use the resource. If another

process requests that resource, the requesting process must be delayed

until the resource has been released.

In this case, every car takes up a place where other cars cannot take up.

Only if the car moves and leaves the place, other cars can take up that place.

**Hold and wait**. A process must be holding at least one resource and

waiting to acquire additional resources that are currently being held by

other processes.

In this case, every car takes up a piece of road and wants to move on the road, but each car can not move on the road because another car takes up the piece of road which is in front of that car.

**No preemption**. Resources cannot be preempted.

In this case, every car can not preempt another car’s place and must wait for the car that is in front of them to leave.

**Circular wait**. A set {*P*0, *P*1, ..., *Pn*} of waiting processes must exist such

that *P*0 is waiting for a resource held by *P*1, *P*1 is waiting for a resource

held by *P*2, ..., *Pn*−1 is waiting for a resource held by *Pn*, and *Pn* is waiting

for a resource held by *P*0.

In the case, there is a circle of cars. The first car C1 wants to move but it should wait for the car C2 that is in front of C1. C2 wants to move but it should waits for the car C3 which is in front of C2. So every car waits for the car that is in front of them to move. The last car will wait for the first to move. So these cars form a circle.

**b.**

1. Make the road into two lanes. This can make two cars share same road, which will break mutual exclusion.

2. Set up traffic lights in the four crossroads, and these four lights can’t be green at the same time. So that no circle of cars occurs.