Exercise 3.46.

Suppose that we implement `test-and-set!` using an ordinary procedure as shown in the text, without attempting to make the operation atomic. Draw a timing diagram like the one in figure 3.29 to demonstrate how the mutex implementation can fail by allowing two processes to acquire the mutex at the same time.

Answer.

The timing diagram in figure 1 illustrates how mutex implementation based on non-atomic `test-and-set!` can fail by allowing two processes to access simultaneously. Here the mutex starts at `false`, a process P_1 acquires the mutex and set it to `true`. Just before P_1 can accomplish its assignment operation, there appeals another one P_2 which, also test the cell to be `false`, and yet both process acquire the mutex finally. We see that the non-atomic implementation of `test-and-set!` causes interleaving in the order of event while two process try to acquire the mutex at the same time. The reason for this anomaly is that by allowing interleaving events, the assumption made by P_2 on the value of the cell has been quite opposite to the fact.

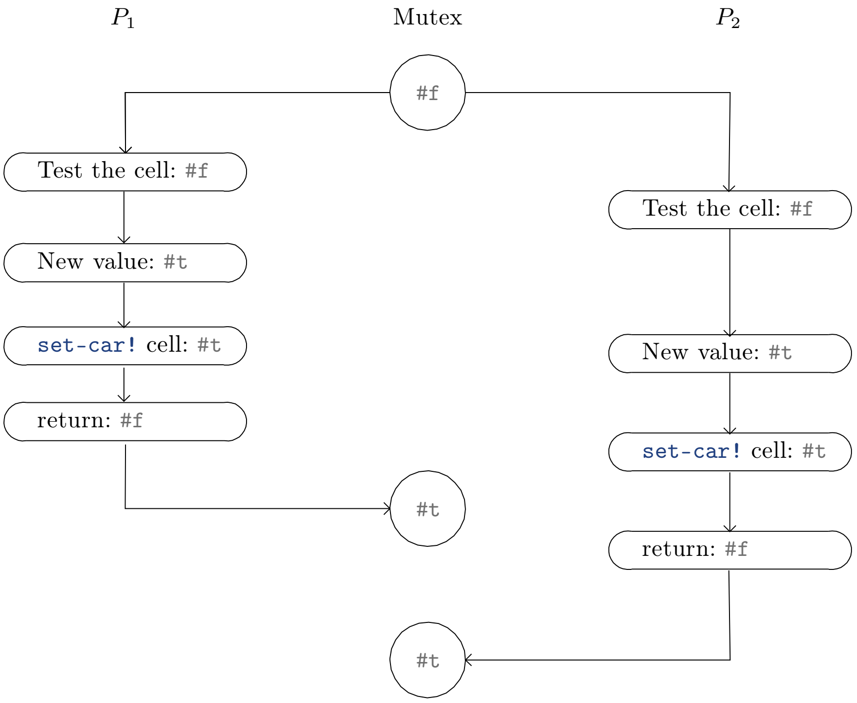

Figure 1. Timing diagram showing how non-atomic implemantation of `test-and-set!` can cause the mutex fail.