TD - 30

Aluno: Vinicius Carloto Carnelocce

1. T1:

```
read_lock (A);
read (A);
read_lock (B);
read (B);
if A = 0 then B := B + 1;
write_lock (B);
write (B);
unlock (A);
unlock (B).
```

   T2:

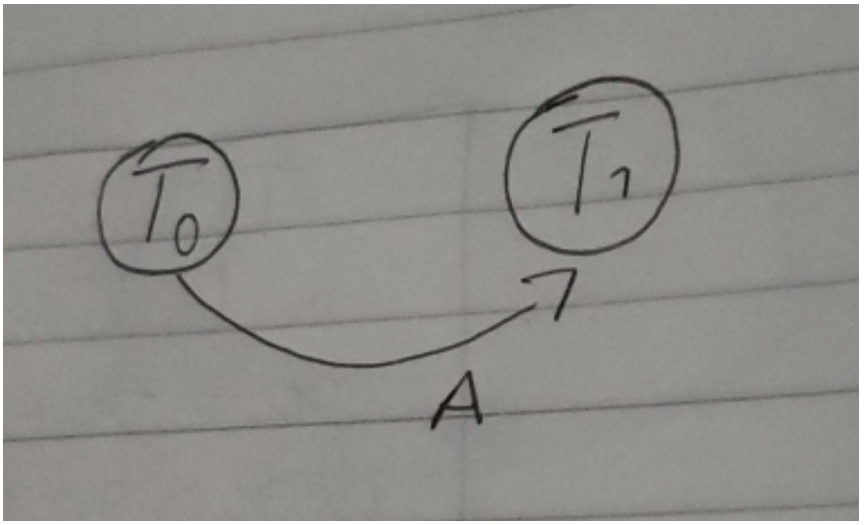
```
read_lock (B);
read (B);
read_lock (A);
read (A);
if B = 0 then A:= A + 1;
write_lock (A);
write (A);
unlock (A);
unlock (B);
```

   Como T1 possui A em read_lock, T2 não conseguirá fazer o write_lock (A). Ao mesmo tempo, T2 possui B em read_lock, impossibilitando que T1 faça write_lock (B). Logo, a adição de bloqueios causa deadlock.

2. Seguem abaixo os schedules com conflito:
   a. gravação e leitura: r2(X); r2(Y); r1(X); r1(Y); w1(X); r2(X); w2(X); w2(Y);
   b. leitura e gravação: r2(X); r2(Y); r1(X); r1(Y); r2(X); w1(X); w2(X); w2(Y);
   c. gravação e gravação: r2(X); r2(Y); r1(X); r1(Y); r2(X); w1(X); w2(X); w2(Y);

   Nos schedule (a), (b) e (c) a transação T1 detém o bloqueio exclusivo de X, impedindo operações de leitura e gravação até que seja confirmado.

3.
   a. A ausência de ciclos no grafo de precedência prova que o schedule é serializável. Segue abaixo:



O schedule seria equivalente é t0 -> t1.

b. Segue abaixo o schedule com bloqueios:

T0	T1
Lock0(A)	
Lock0(B)	
Read0(A)	
Write0(A)	
Unlock0(A)	
	Lock1(A)
	Read1(A)
	Write1(A)
	Unlock1(A)
	C1
Read0(B)	
Write0(B)	
Unlock0(B)	
C0	

O schedule acima é equivalente ao schedule serial t0 -> t1;

c. Segue abaixo o schedule com bloqueios sem 2PL:

T0	T1
Lock0(A)	
Read0(A)	
Write0(A)	
Unlock0(A)	
	Lock1(A)

Read1(A)
Write1(A)
Unlock1(A)
C1

Lock0(B)
Read0(B)
Write0(B)
Unlock0(B)
C0

4.

a. O schedule não é serializável pois os conflitos envolvendo X configuram um ciclo entre T1 e T2 no grafo precedência.
b. O 2PL estrito não é aplicável a esse schedule, pois ambas transação possuem bloqueio exclusivo de X, causando deadlock.