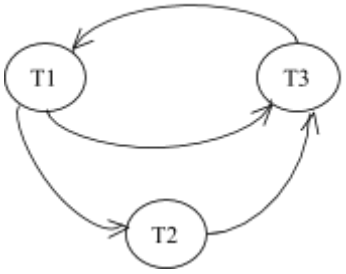
Lista de Exercícios VII
Gerenciamento de Transações, Controle de Concorrência e Recuperação de Falhas

João Batista de Sousa Paula, 11911BCC008
John Vitor da Silva Cunha, 11821BCC005
Laís Saloum Deghaide, 11821BCC001
Vinícius Calixto Rocha, 11911BCC039
Yan Stivaletti e Souza, 11821BCC002

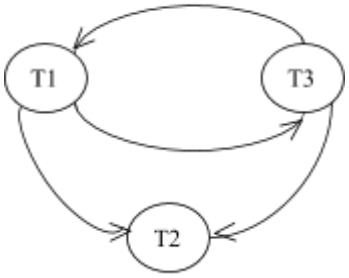
1.

a) r1(x) < w3(x)
r3(x) < w1(x)
w1(x) < r2(x)
r2(x) < w3(x)
w1(x) < w3(x)



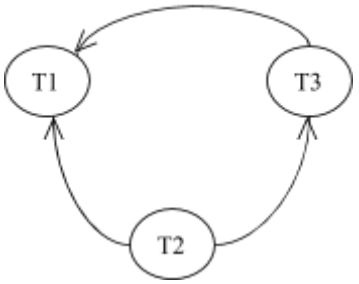
Não é serializável de conflito, pois possui ciclos em seu histórico.

b) r1(x) < w3(x)
r3(x) < w1(x)
w3(x) < w1(x)
w3(x) < r2(x)
w1(x) < r2(x)



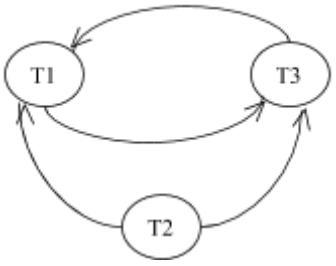
Não é serializável.

c) r3(x) < w1(x)
r2(x) < w3(x)
r2(x) < w1(x)
w3(x) < r1(x)



É serializável. O schedule serial equivalente é: r2(x), r3(x), w3(x), r1(x), w1(x).

d) r3(x) < w1(x)
r2(x) < w3(x)
r2(x) < w1(x)
r1(x) < w3(x)
w3(x) < w1(x)

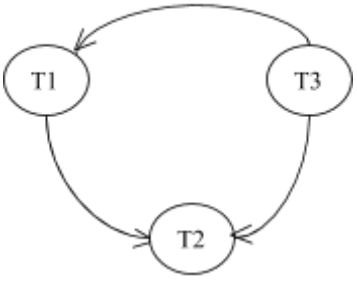


Não é serializável.

2.

S1:

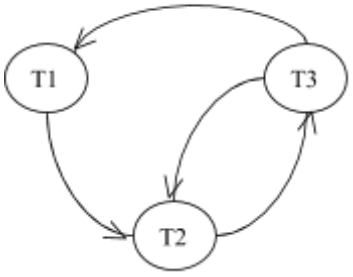
r3(x) < w1(x)
w3(y) < r2(y)
r1(z) < w2(z)
r3(y) < w2(y)



É serializável. O schedule serial equivalente é: r3(x), w3(x), r3(y), w3(y), r1(x), w1(x), r1(z), w1(z), r2(y), w2(y), r2(z), w2(z).

S2:

r3(x) < w1(x)
r1(z) < w2(z)
r2(y) < w3(y)
r3(y) < w2(y)
w3(y) < w2(y)



Não é serializável, pois possui ciclo.

3. S3: É um schedule recuperável (pois a confirmação é dada logo após a leitura), evita rollback em cascata (todas as leituras que ocorreram após a escrita foram de itens já confirmados) e não é escrito (há múltiplas transações ocorrendo simultaneamente).

   S4: É um schedule não recuperável (pois há leitura antes de dados não confirmados), não evita rollback em cascata (as leituras ocorrem antes da escrita de itens não confirmados) e não é escrito (pois há múltiplas transações ocorrendo simultaneamente).

   S5: É um schedule recuperável (pois pois a confirmação é dada logo após as leituras leituras de todos os itens), não evita rollback em cascata (simplesmente pois não há leitura após a escrita de um item) e não é escrito (há múltiplas transações ocorrendo simultaneamente).

4. O bloqueio Binário possui dois estados, bloqueado e desbloqueado, e impõe a exclusão mútua do item; se uma operação de bloqueio ou desbloqueio de um dado X for iniciada, nenhum entrelaçamento é permitido até que a operação em questão termine ou a transação espere, ou seja, o comando de espera coloca a transação em uma fila de espera pelo item X até que ele seja desbloqueado. Já um esquema de bloqueio múltiplo permite que um item de dado seja acessado por mais de uma transação para leitura, requerendo a implementação de uma tabela de bloqueios e uma fila de espera. os bloqueios exclusivos são preferíveis pois garantem melhor fluxo e otimização no acesso aos dados, podendo acessar múltiplas tabelas ao invés de esperar o desbloqueio do dado em questão.

5. Para as operações de inserção e exclusão é necessário o bloqueio exclusivo, write_lock(x), pois é ele que permite que aquele dado seja tratado exclusivamente pela transação impedindo que outra transação use aquele dado até ele ser desbloqueado, unlock(x).

6. O t2 e t3 serão ignorados por não terem atingido seus pontos de confirmações
só fará diferença no t4 é refeito porque seu ponto de confirmação depois do último checkpoint do sistema.

7. Não, pois a Transação T1 ainda termina em seu ponto de confirmação sem realizar leituras de itens de outras transações, não precisando ser restaurada, a T2 além de ter realizado a leitura do item B que é escrito por T3 não atingiu seu ponto de confirmação assim como a T3 necessitando serem revertidas.
Por isso, a T2 e T3 devem serem revertidas ao contrário de T1, não realizando alterações no processo de recuperação listado.