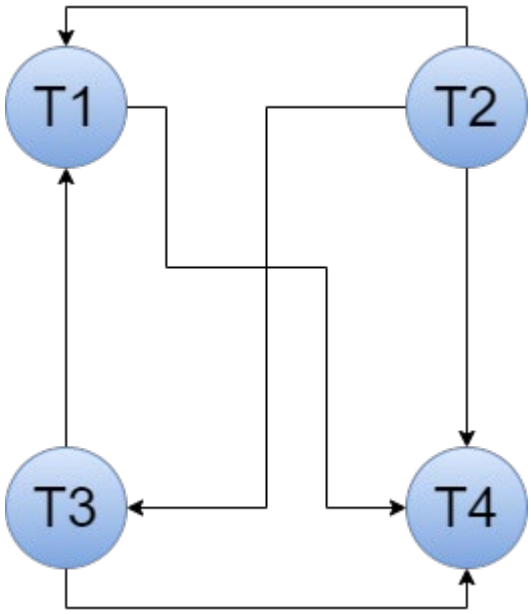
Nome: Davi Augusto Neves Leite

RA: 191027383

Resolução – Segunda Lista de Exercícios – Banco de Dados II

1)

a) Grafo de Precedência **P(S)**:



Grafo de Precedência Rotulado **PR(S)**:

---

b) Como o grafo de precedência P(S) não possui ciclos, então a escala S é serializável no conflito.

c) Como a escala S é serializável no conflito, então ela também é serializável na visão (por teorema).

d) Para saber se a escala S é legal segundo o protocolo 2PL, é necessário inserir os bloqueios e desbloqueios dos dados nas transações, conforme o protocolo implica. Dessa forma, a escala S ficaria:

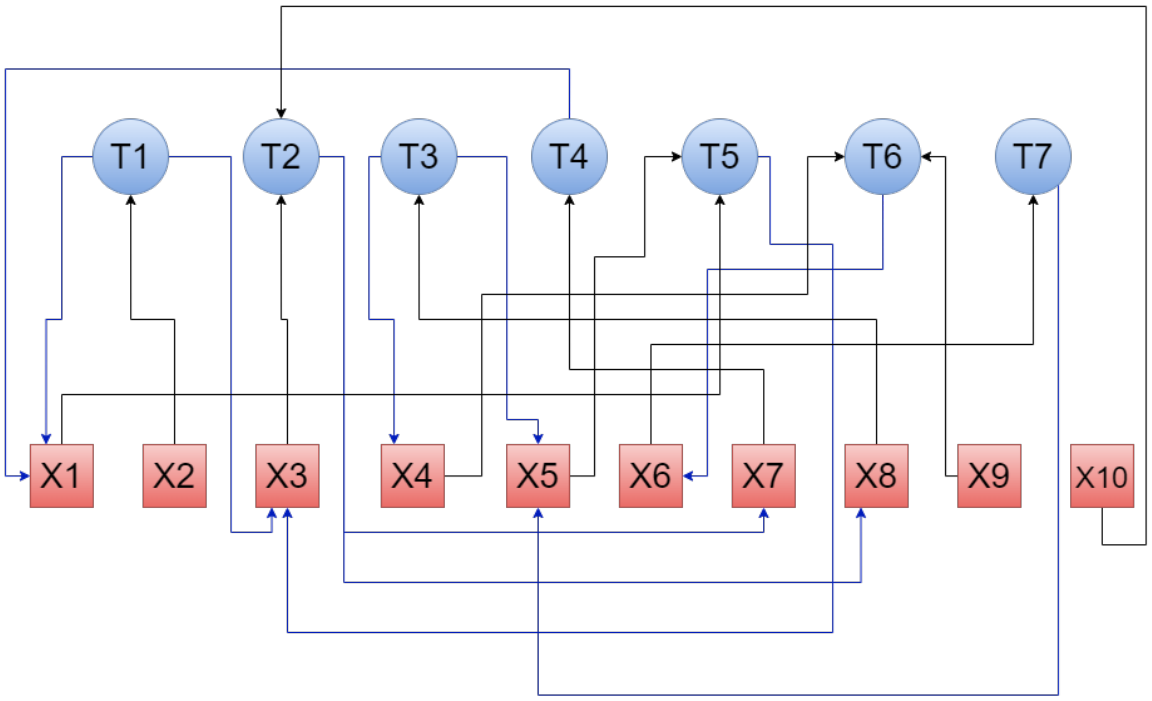
S-2PL			
T1	**T2**	**T3**	**T4**
	LX(A)		
	R(A)		
	LX(B)		
	R(B)		
	W(A)		

	U(A)		
		LX(A)	
		R(A)	
	W(B)		
	U(B)		
LX(B)			
		W(A)	
		U(A)	
W(B)			
U(B)			
			LX(B)
			R(B)
LX(A)			
R(A)			
			W(B)
			U(B)
LX(C)			
W(A)			
U(A)			
			LX(A)
			R(A)
			W(A)
			U(A)
W(C)			
U(C)			

Dessa forma, a escala S é legal segundo o protocolo 2PL.

e) Com relação ao protocolo TO, a escala S não é legal uma vez que possui a operação de leitura do dado A na transação T3 após a escrita do mesmo dado na transação T2. Dessa forma, não é possível atribuir a ordenação de TimeStamp.

2) Conforme o grafo de espera abaixo, não há *deadlock*.



3)

Código Oracle SQL para criação da tabela FUNCIONÁRIOS e de duas transações T1 e T2:

```
rem ************ Nome: Davi Augusto Neves Leite ******************
rem ************ RA: 191027383                   ******************

rem *************** SGBD: Oracle SQL ***********************

rem Pré-requisitos: criação e população da tabela 'Funcionários'

create table Funcionario(
  rg number(2) generated by default on null as identity primary key,
  nome varchar2(50) not null,
  endereco varchar2(100) not null,
  salario number(10,2) not null,
  departamento varchar2(50) not null,
  funcao varchar2(50) not null
);

rem Transação 1: inserir e retornar os novos dados da tabela
Funcionário (operação de escrita e leitura)
set transaction read write;
```

```
insert into Funcionario (nome, endereco, salario, departamento,
funcao)
values ('Davi', 'Rua Não Sei', ROUND(DBMS_RANDOM.VALUE(1, 10000)),
'Computação', 'Aluno');

select * from Funcionario;

commit;

rem Transação 2: atualizar e retornar os novos dados da tabela
Funcionário (operação de escrita e leitura)
set transaction read write;

select * from Funcionario;

update Funcionario
set
    nome = 'Nilceu Marana',
    endereco = 'Rua Continuo Não Sabendo',
    salario = ROUND(DBMS_RANDOM.VALUE(1, 10000)),
    funcao = 'Professor'
where exists
    (select rg from Funcionario
     where nome = 'Davi');

select * from Funcionario;

rollback;
```

Diante disso, pode-se propor a seguinte escala de execução para as transações T1 e T2 de tal forma que acarrete o *deadlock* (ciclo de alocação e espera de recursos):

SD-T1T2	
T1	**T2**
LX(A)	
	LS(B)
W(A)	
R(A)	
	R(B)
	LX(A)
LX(B)	