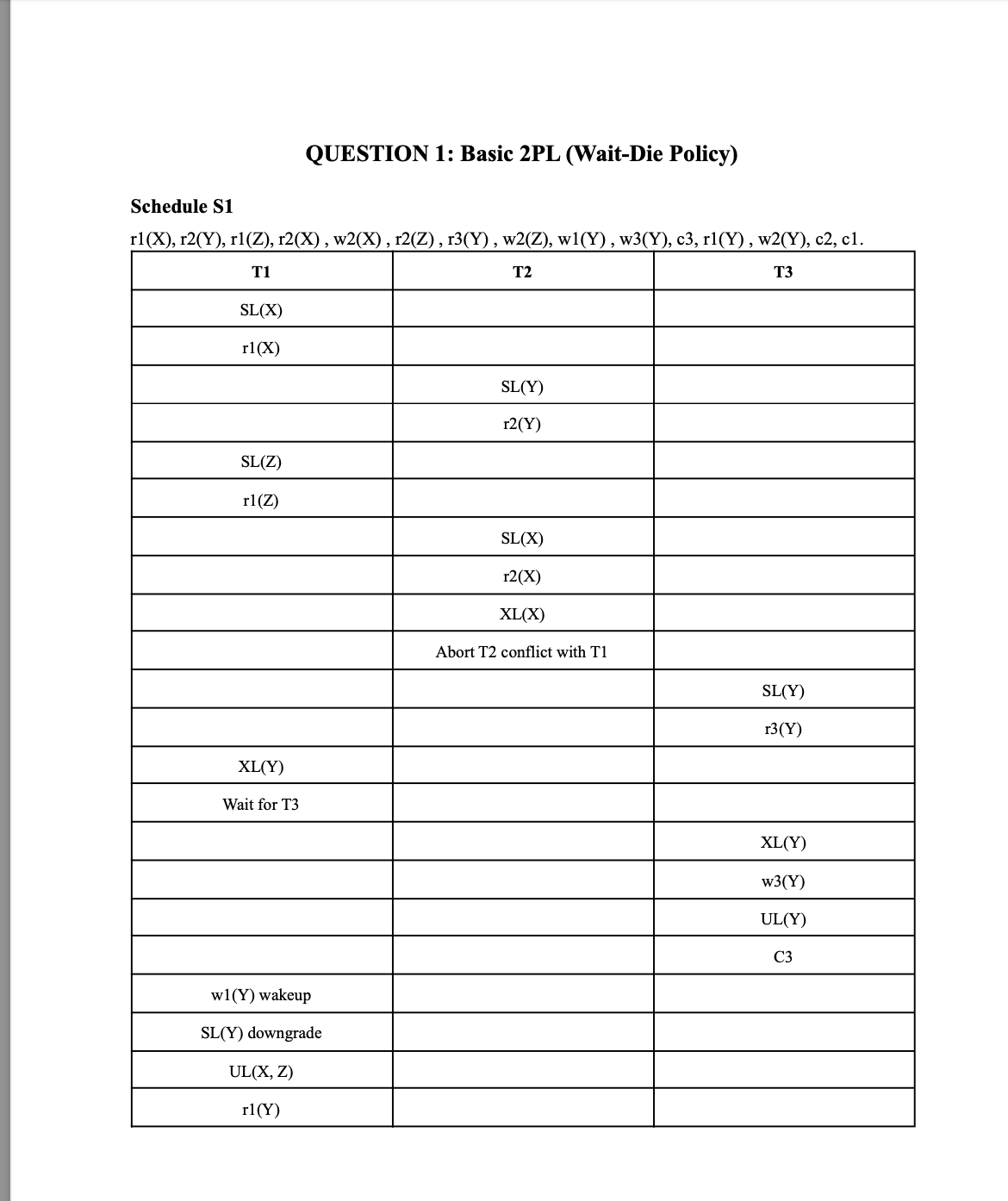
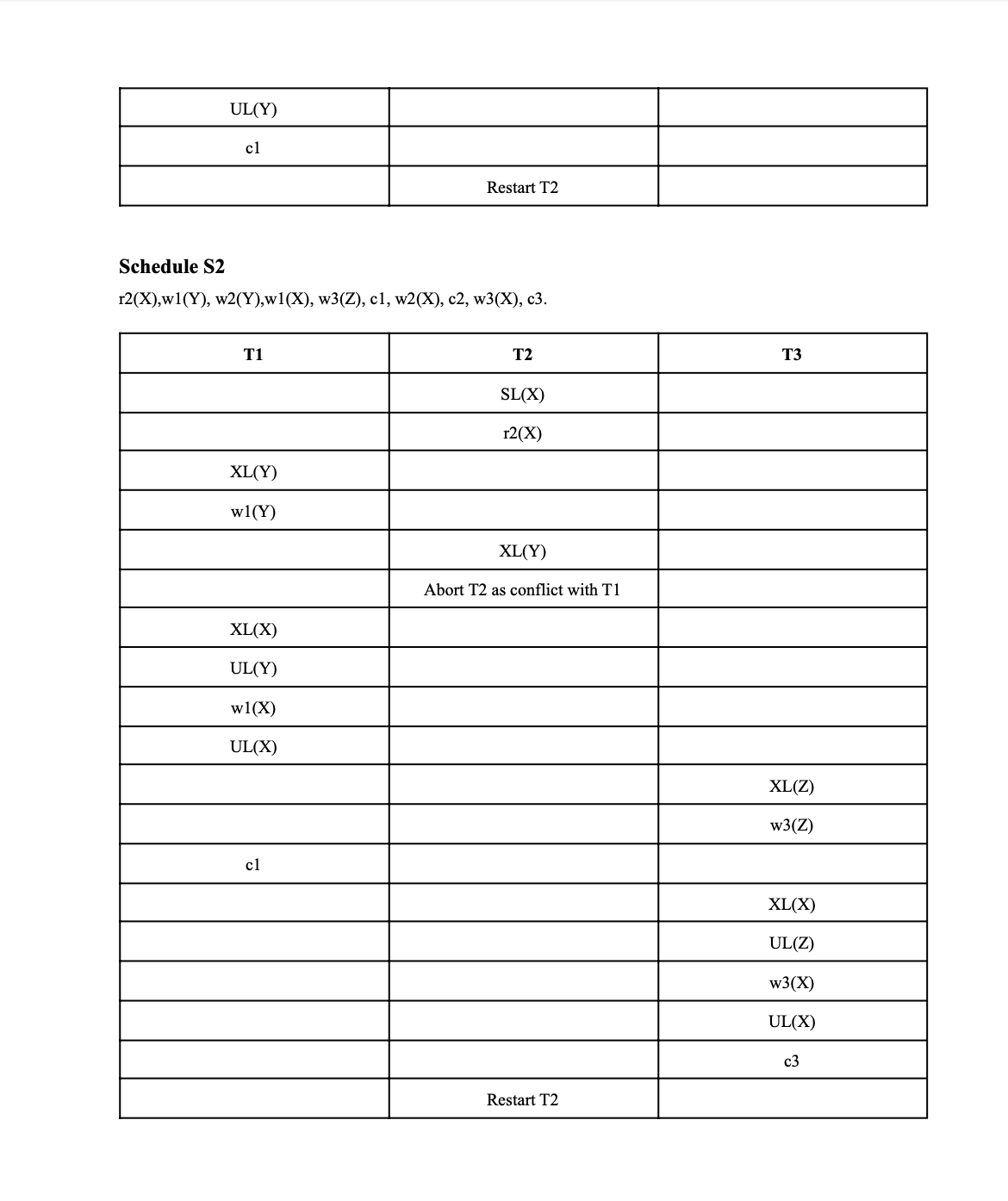
**Advance Database Concepts**

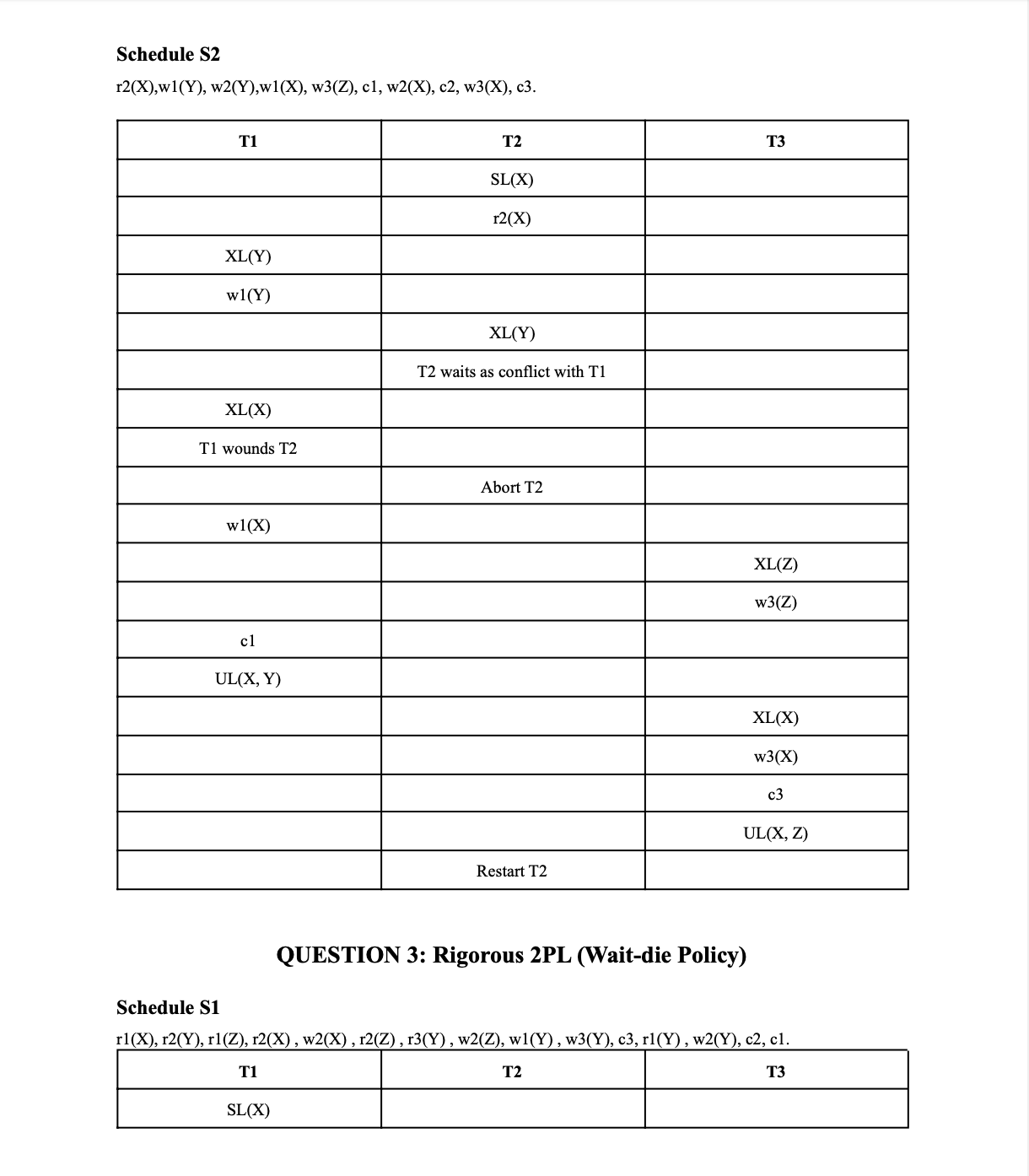
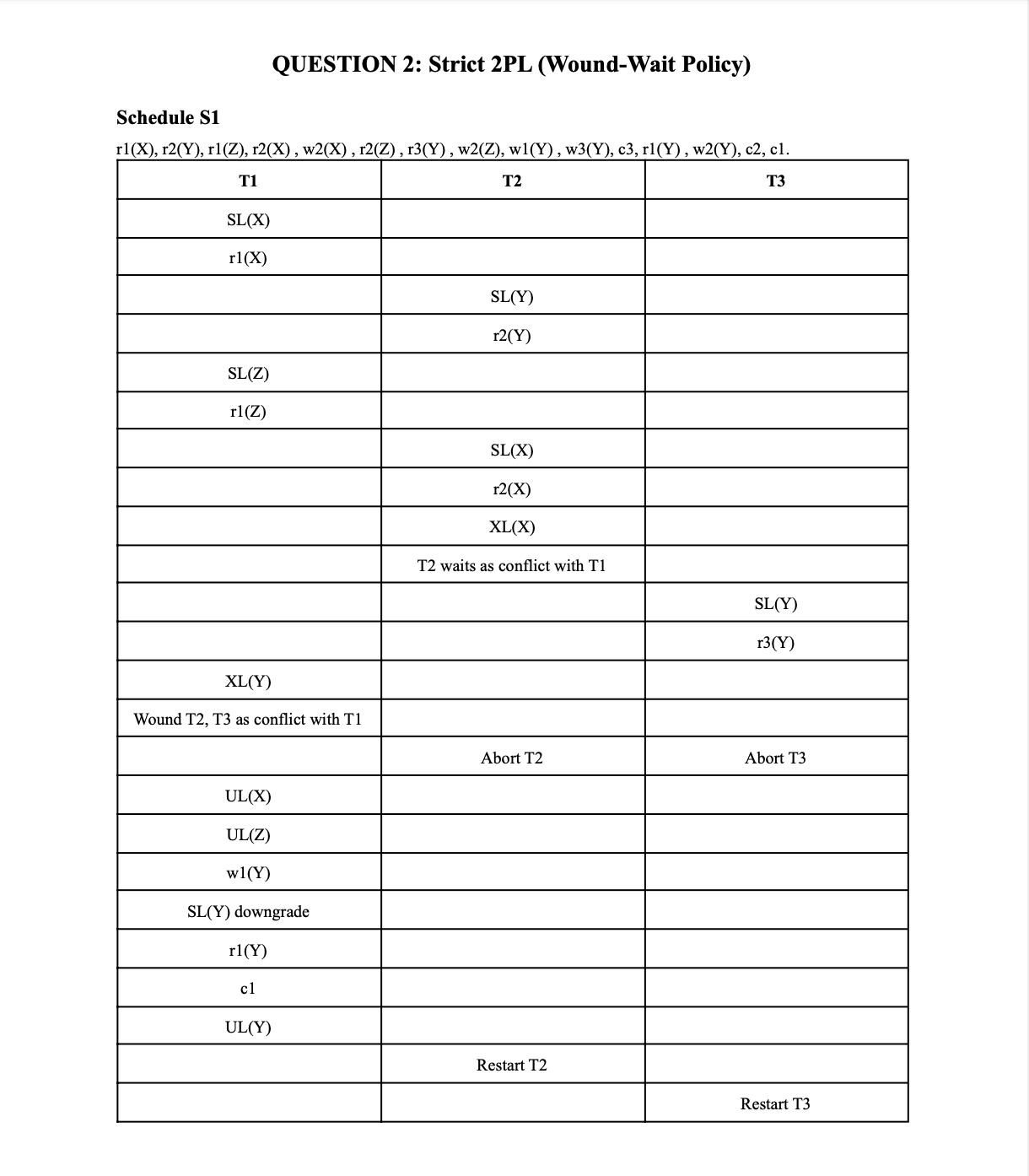
**Assignment # 01**

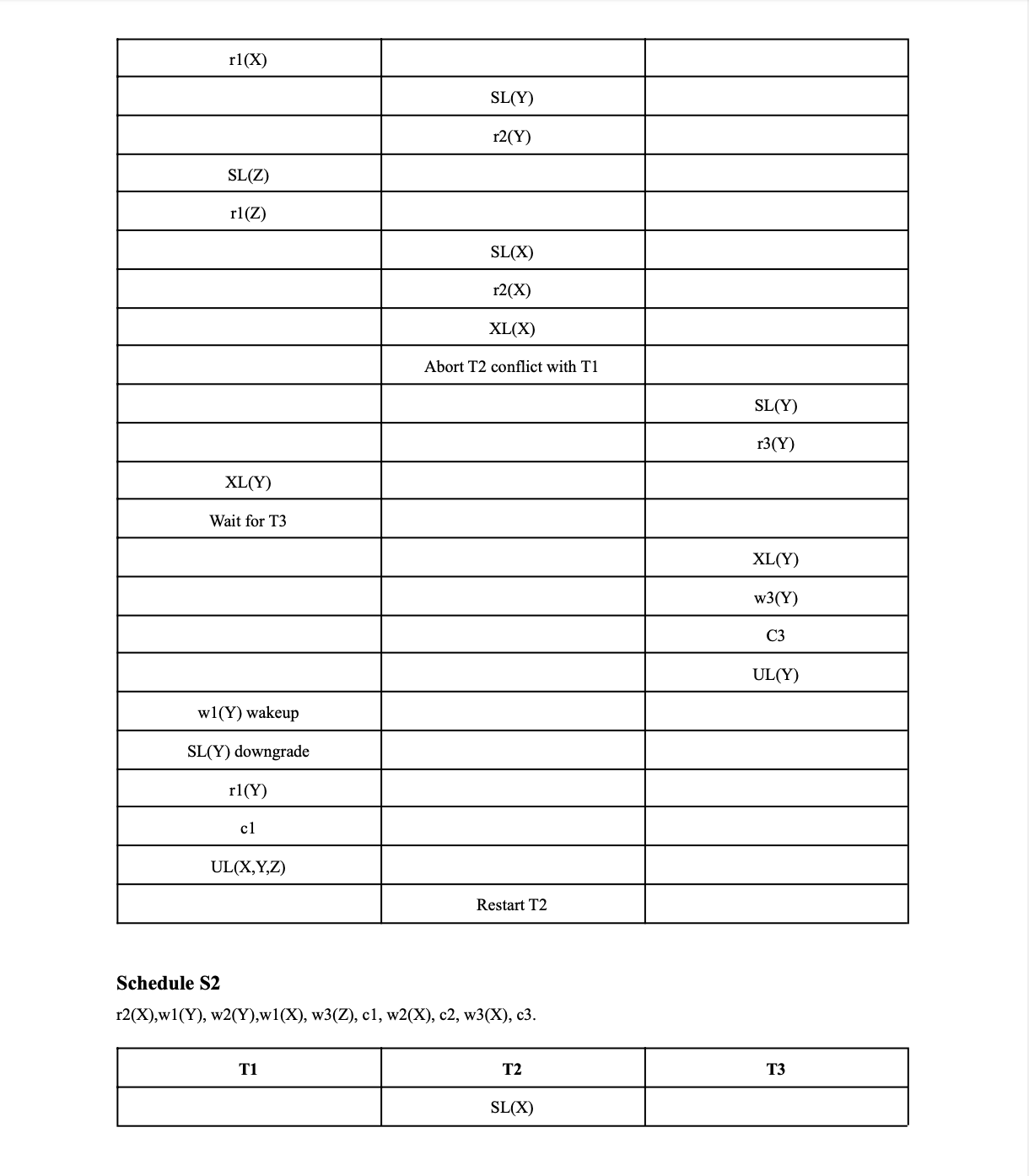
**Solution**

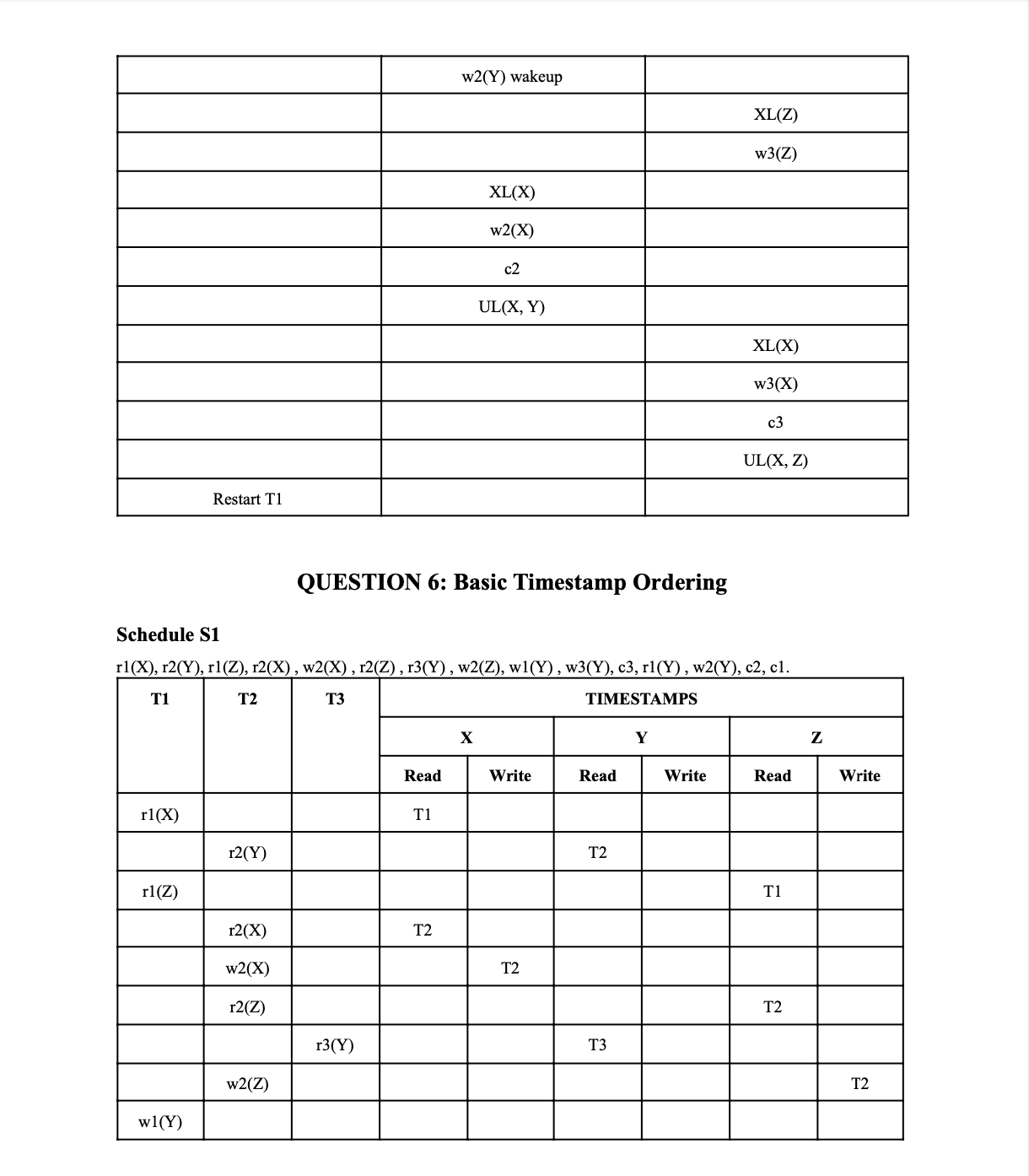
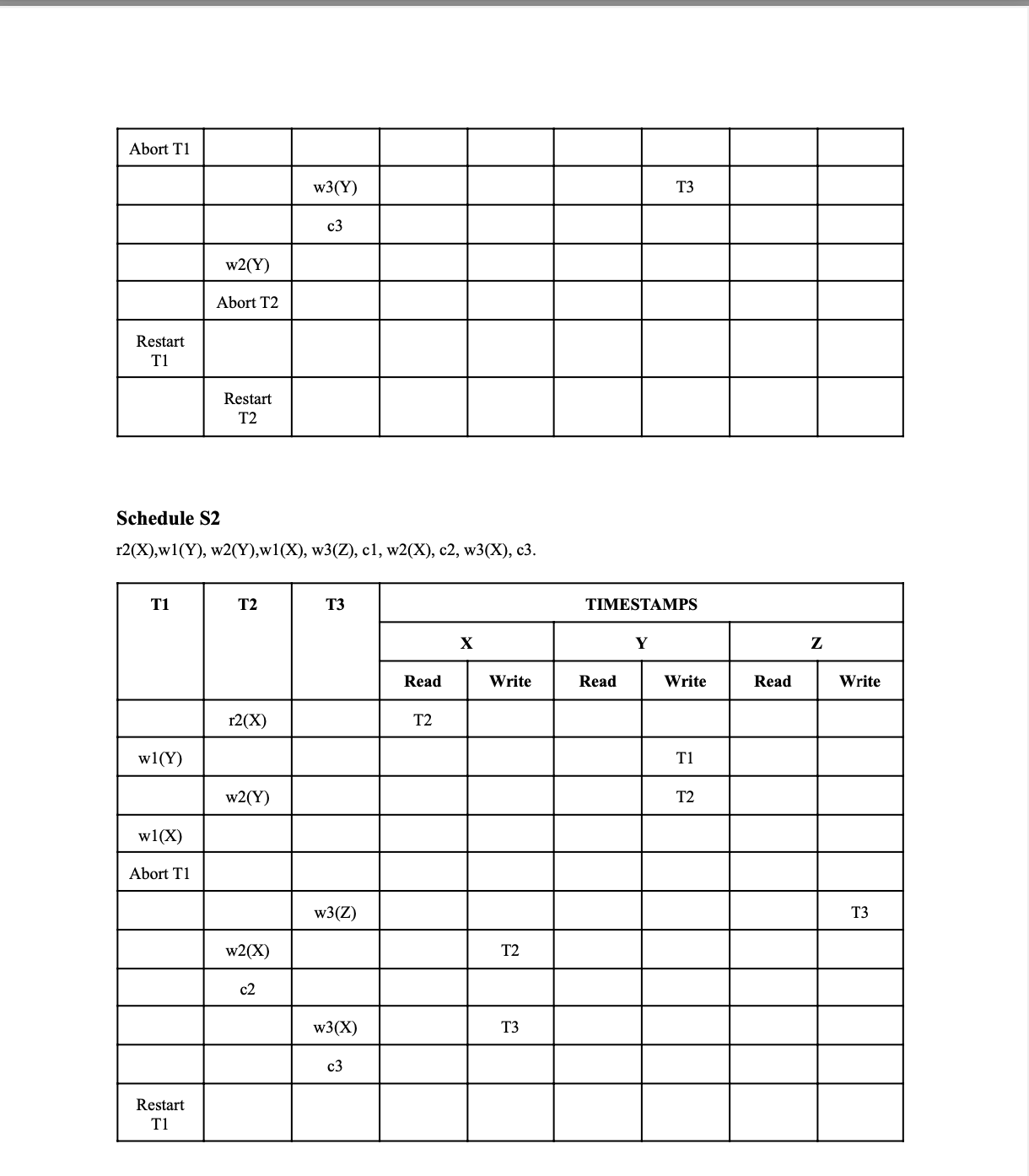
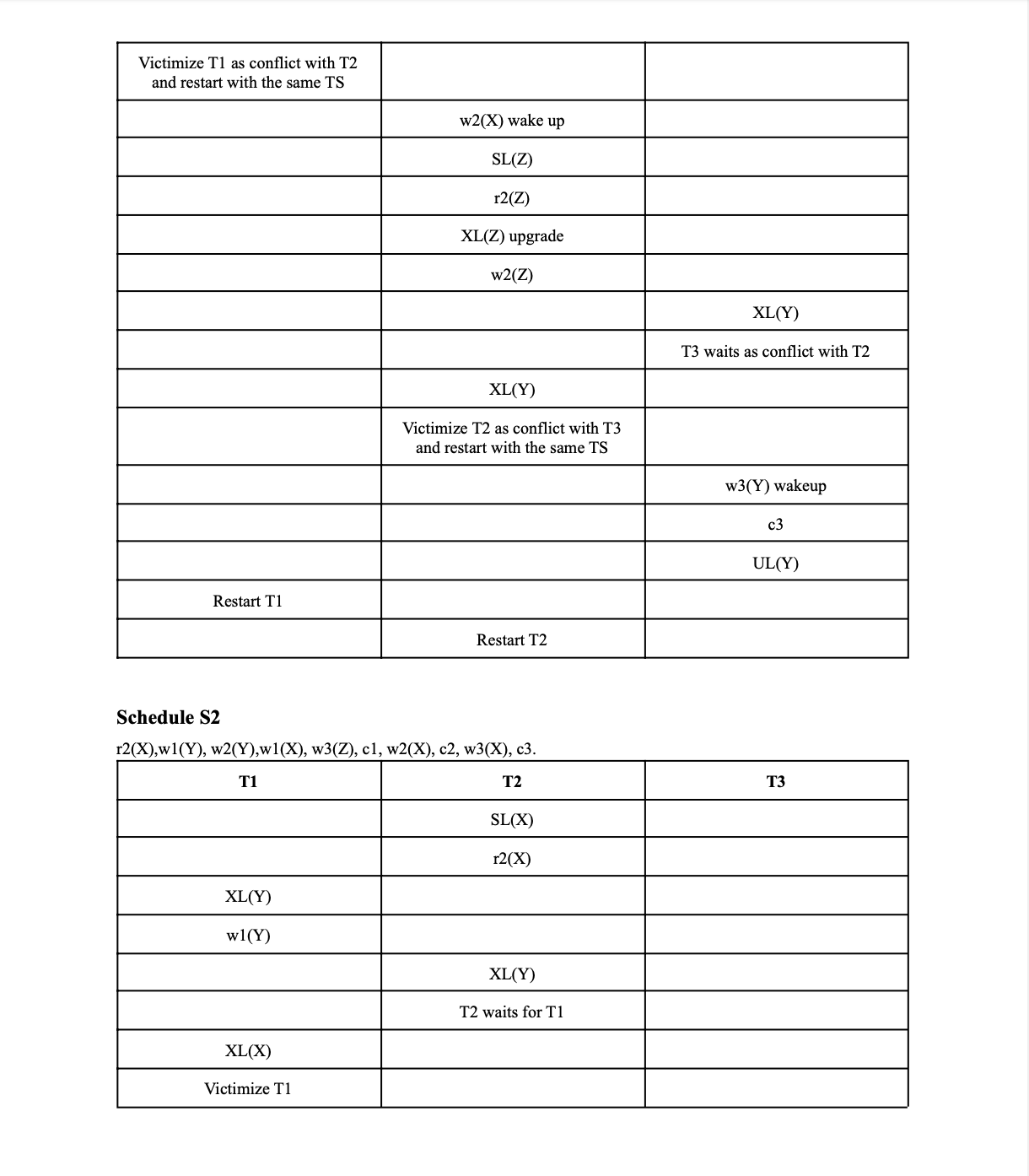
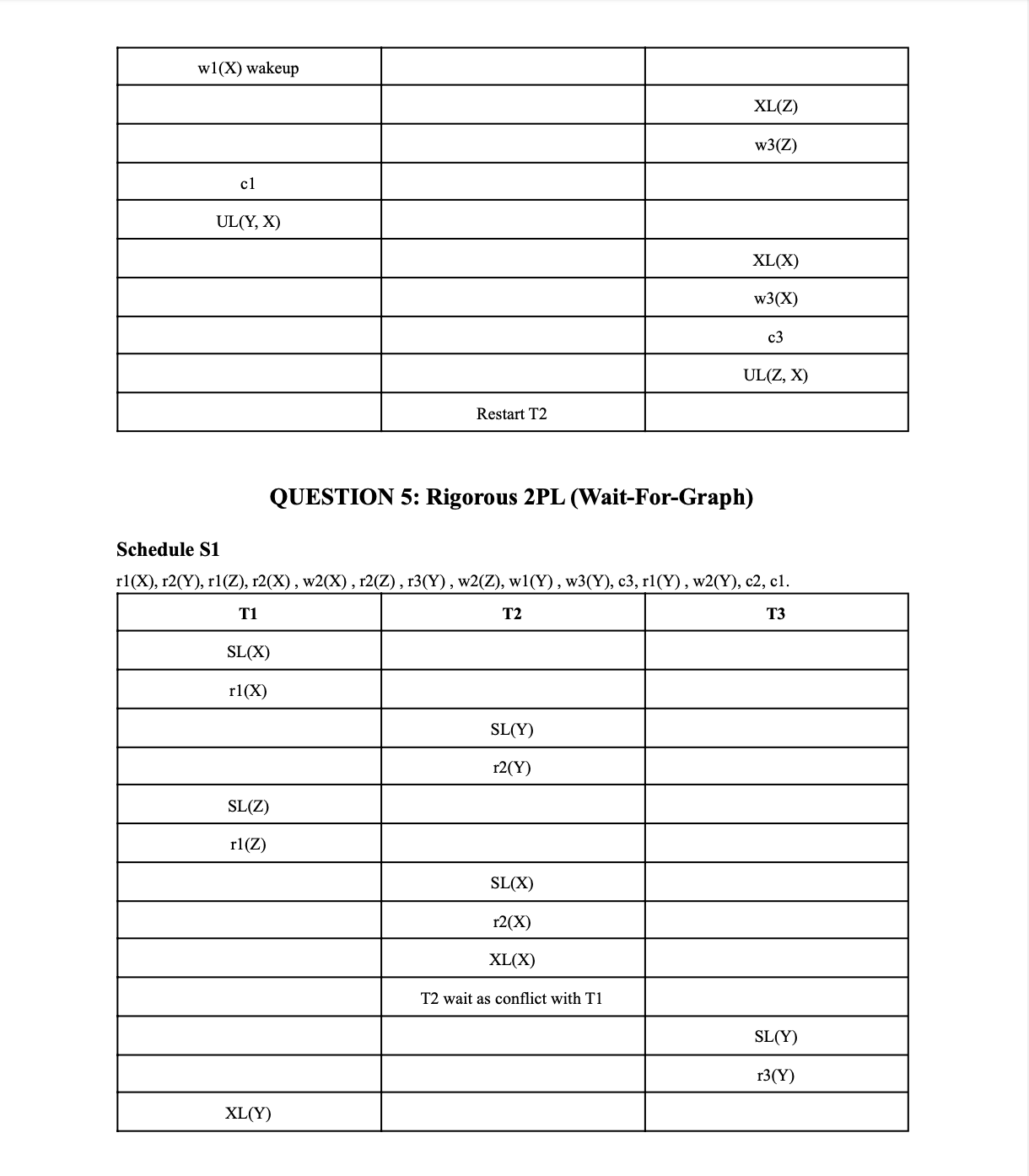
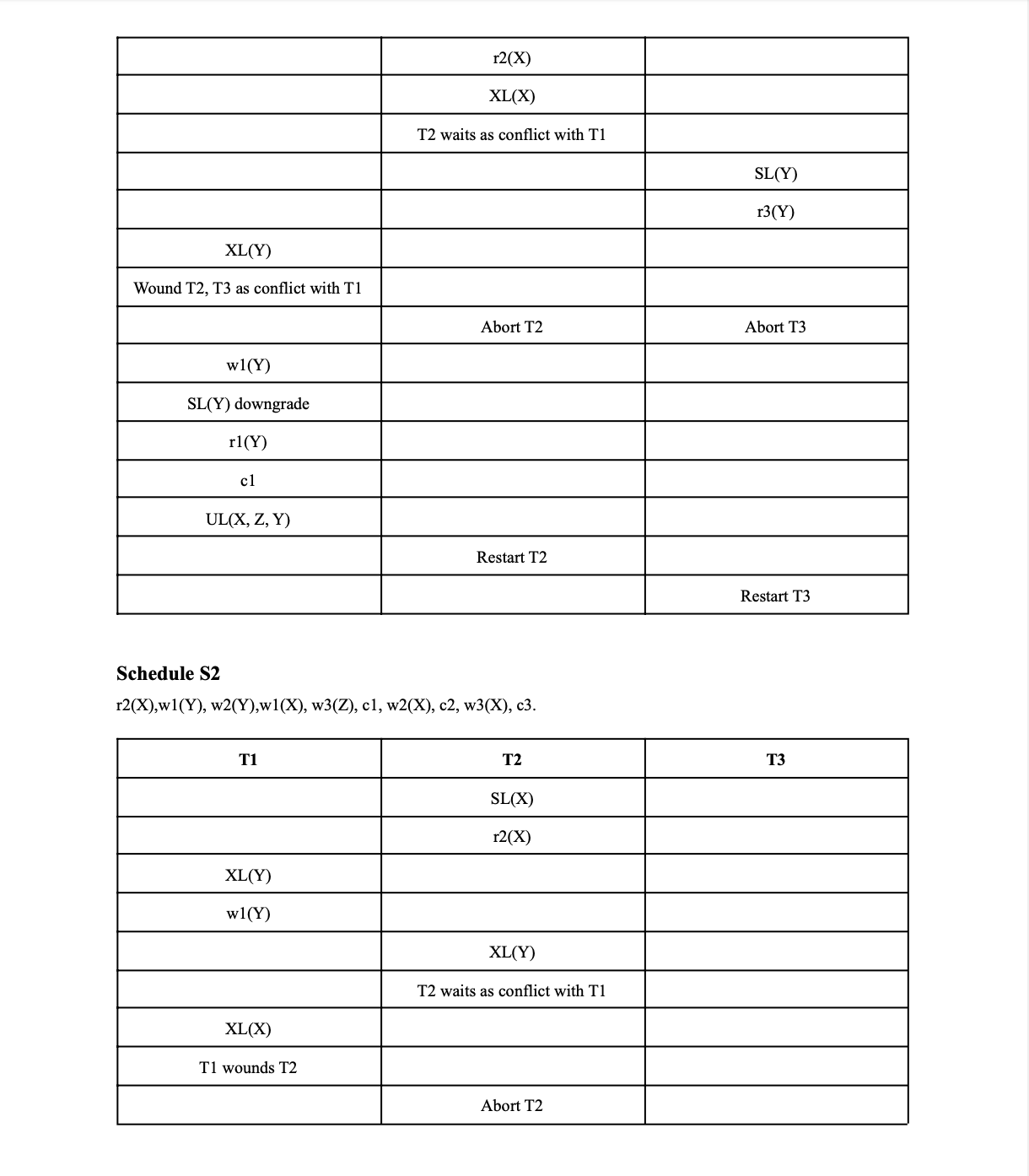
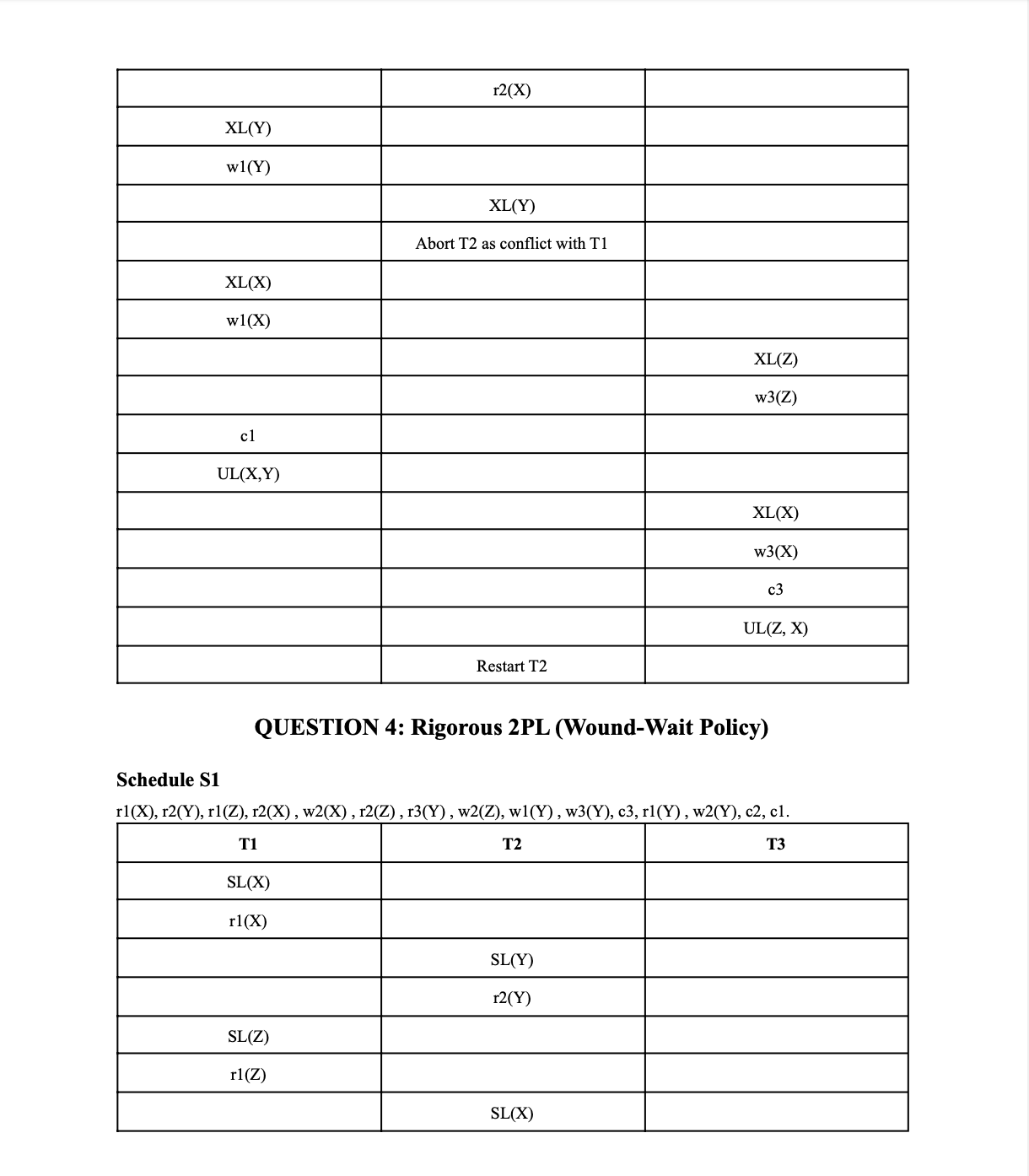


‘











Q7: Strict Timestamp Ordering (TO) protocol

S1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | X RD | X WR | Y RD | Y WR | Z RD | Z WR |
| r1(X)  r1(Z)  W1(Y)  Abort due to T3 | r2(Y)  r2(X)  w2(X)  r2(Z)  w2(Z)  w2(Y)  Abort Due to T3 | r3(Y)  w3(Y)  c3 | T1  T2 | T2 | T2  T3 | T3 | T1  T2 | T2 |

S2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | X RD | X WR | Y RD | Y WR | Z RD | Z WR |
| w1(Y)  w1(X)  Abort | r2(X)  w2(Y)  Wait for T1 to commit or abort  w2(Y)  w2(X)  c2 | w3(Z)  w3(X)  c3 | T2 | T2  T3 |  | T1  T2 |  | T3 |

Q8: Thomas Write Rule Timestamp Ordering (TO) protocol

S1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | X RD | X WR | Y RD | Y WR | Z RD | Z WR |
| r1(X)  r1(Z)  w1(Y)  Abort due to T3 | r2(Y)  r2(X)  w2(X)  r2(Z)  w2(Z)  w2(Y)  Abort Due to T3 | r3(Y)  w3(Y)  c3 | T1  T2 | T2 | T2  T3 | T3 | T1  T2 | T2 |

S2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | X RD | X WR | Y RD | Y WR | Z RD | Z WR |
| w1(Y)  w1(X)  Abort due to T2 | r2(X)  w2(Y)  w2(X)  c2 | w3(Z)  w3(X)  c3 | T2 | T2  T3 |  | T1  T2 |  | T3 |

Q9: MultiVersion Timestamp Ordering (TO) protocol

S1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | X RD | X WR | Y RD | Y WR | Z RD | Z WR |
| r1(X)  r1(Z)  w1(Y)  Abort due to T3 | r2(Y)  r2(X)  w2(X)  r2(Z)  w2(Z)  w2(Y)  Abort Due to T3 | r3(Y)  w3(Y)  c3 | {T1}  {T2}  {T2} | T2 | {T2}  {T3}  {T3} | T3 | {T1}  {T2}  {T2} | T2 |

S2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | X RD | X WR | Y RD | Y WR | Z RD | Z WR |
| w1(Y)  w1(X)  Abort due to T2 | r2(X)  w2(Y)  w2(X)  c2 | w3(Z)  w3(X)  c3 | {T2}  {T2}  {T3} | T2  T2, T3 | {T1}  {T2} | T1  T1,T2 | {T3} | T3 |

Q10.Validation (Optimistic) Concurrency Control Technique (Use defer the validation

until a later time when the conflicting transactions have finished)

S1:

T3 Committed first , hence its validation starts first:

Backward Validation(compares read set with old write set): {Y} ∩ {} = {} → success

Forward Validation(compares write set with active read set): {Y} ∩ {X,Y,Z} = {Y} → failed

Since forward validation failed so delaying validation of T3

Now check for T2:

Backward Validation(compares read set with old write set): {X,Y,Z} ∩ {} = {} → success

Forward Validation(compares write set with active read set): {X,Y,Z} ∩ {X,Y,Z} = {X,Y,Z} → failed

Since forward validation failed so delaying validation of T2

Now check for T1:

Backward Validation(compares read set with old write set): {X,Y,Z} ∩ {} = {} → success

Forward Validation(compares write set with active read set): {Y} ∩ {X,Y,Z} = {X,Y,Z} → failed

All validations failed

S2:

T1 Committed first , hence its validation starts first:

Backward Validation(compares read set with old write set): {} ∩ {} = {} → success

Forward Validation(compares write set with active read set): {X,Y} ∩ {X} = {X} → failed

Since forward validation failed so delaying validation of T1

Now check for T2:

Backward Validation(compares read set with old write set): {X} ∩ {} = {} → success

Forward Validation(compares write set with active read set): {X,Y} ∩ {} = {} → success

So validation of T2 is successful.

Now check for T3:

Backward Validation(compares read set with old write set): {} ∩ {X,Y} = {} → success

Forward Validation(compares write set with active read set): {Z,X} ∩ {} = {} → success

So validation of T3 is successful.

Validation of T1:

Backward Validation(compares read set with old write set): {} ∩ {X,Y,Z} = {} → success

Forward Validation(compares write set with active read set): {X,Y} ∩ {} = {X} → success

So validation of T1 is successful.