Assignment 2

Question 2 (4 marks)

1) (2 marks)

**Step 1 Reduce Right Side:**

\[ F' = \{ AB \rightarrow C, AB \rightarrow D, E \rightarrow D, \ ABC \rightarrow D, \ ABC \rightarrow E, E \rightarrow A, E \rightarrow B, D \rightarrow A, D \rightarrow G, ACD \rightarrow B, ACD \rightarrow E \} \]

**Step 2 Reduce Left Side:**

For \( ABC \rightarrow D \), \( AB \rightarrow D \) is inferred by \( F' \). Hence, \( ABC \rightarrow D \) is replaced by \( AB \rightarrow D \).

Similarly, we can replace \( ABC \rightarrow E \) with \( AB \rightarrow E \), \( ACD \rightarrow B \) with \( CD \rightarrow B \), \( ACD \rightarrow E \) with \( CD \rightarrow E \).

**Step 3 Reduce Redundancy:**

\( \{ E \}^+_{F' \setminus \{ E \rightarrow D \}} = \{ A, B, C, D, E, G \} \), so \( E \rightarrow D \) is redundant. Thus, we remove it from \( F' \).

Similarly, we can remove \( AB \rightarrow E \) and \( CD \rightarrow B \).

Thus, \( F_m = \{ AB \rightarrow C, D \rightarrow A, D \rightarrow G, E \rightarrow B, AB \rightarrow D, E \rightarrow A, CD \rightarrow E \} \)

This is a sample solution.

2) (2 marks)

For \( F_m = \{ AB \rightarrow C, D \rightarrow A, D \rightarrow G, E \rightarrow B, AB \rightarrow D, E \rightarrow A, CD \rightarrow E \} \):

- From \( AB \rightarrow C, AB \rightarrow D \), derive \( R_1 \{ A, B, C, D \} \)
- From \( D \rightarrow A, D \rightarrow G \), derive \( R_2 \{ A, D, G \} \)
- From \( E \rightarrow B, E \rightarrow A \), derive \( R_3 \{ A, B, E \} \)
- From \( CD \rightarrow E \), derive \( R_4 \{ C, D, E \} \)

None of the relation schemas contains a key of \( R \), add one relation schema \( R_5 \{ E, H \} \)
Question 3 (8 marks)

1) (2 mark)
T1, T2: redo
T3: undo

2) (2 mark)
T2: redo
T3: undo

3) (2 marks)
Yes. There is no cycle in its schedule graph:

4) (2 marks)
There is no way to construct a schedule whose wait-for graph contains cycles.
We have T1 and T3 read and write on X, we have potential to make T1 wait-for T3 or T3 wait-for T1.
We have T2 and T3 read and write on Y, we have potential to make T2 wait-for T3 or T3 wait-for T2.
If we make T1 wait-for T3, we cannot make T3 wait-for T1 directly or through T2.
If we make T3 wait-for T1, we cannot make T1 wait-for T3 directly or through T2.

If we make T2 wait-for T3, we cannot make T3 wait-for T2 directly or through T1.

If we make T3 wait-for T2, we cannot make T2 wait-for T3 directly or through T1.