Computer Science 143, Homework 5

Michael Wu
UID: 404751542

June 8th, 2018

Problem 1

a) Yes. Each transaction does not overlap with any other transaction.

b) Yes it is conflict serializable, as it is already serial. The order of the transactions are currently $T_3 \to T_1 \to T_2 \to T_4$. Note that this schedule is conflict equivalent to $T_3 \to T_1 \to T_4 \to T_2$.

Problem 2

a) We have the following execution schedules.

$$T_1 \to T_2 \to T_3$$
$$T_1 \to T_3 \to T_2$$
$$T_2 \to T_1 \to T_3$$
$$T_2 \to T_3 \to T_1$$
$$T_3 \to T_1 \to T_2$$
$$T_3 \to T_2 \to T_1$$

These yield the following values in order: 30, 30, 42, 36, 25, 37.

b) Since an update statement is both a read and a write, the transaction T_2 will grab a lock on both rows in the table. It then releases it at the end of the transaction. So T_3 cannot write in between T_2, and T_2 and T_3 must

be serializable with respect to each other. However, the transaction T_1 can read at any time. Let $T_2 = AB$ and $T_3 = CD$, where A, B, C, and D are the atomic statements that make up T_2 and T_3. Then we have the following possible execution schedules.

Schedule	Output
T_1ABCD	30
AT_1BCD	32
ABT_1CD	42
$ABCT_1D$	52
$ABCDT_1$	36
T_1CDAB	30
CT_1DAB	40
CDT_1AB	25
$CDAT_1B$	27
$CDABT_1$	37