***Algorithm to test for lossless join:***

*1. Construct an m by n table, S, with a column for each of the n attributes in R and a row for each of the m relations in the decomposition.*

*2. For each cell S(i,j) of S,*

*if the attribute for the column, Aj, is in the relation for the row, Ri,*

*then place the symbol a(j) in the cell*

*else place the symbol b(i,j) there*

*3. Repeat the following process until no more changes can be made to S:*

*for each FD X → Y in F*

*for all rows in S that have the same symbols in the columns corresponding to the attributes of X, make the symbols for the columns that represent attributes of Y equal by the following rule:*

*if any row has an a value, a(j), then set the value of that column in all the other rows equal to a(j)*

*if no row has an a value, then pick any one of the b values, say b(i,j), and set all the other rows equal to b(i,j)*

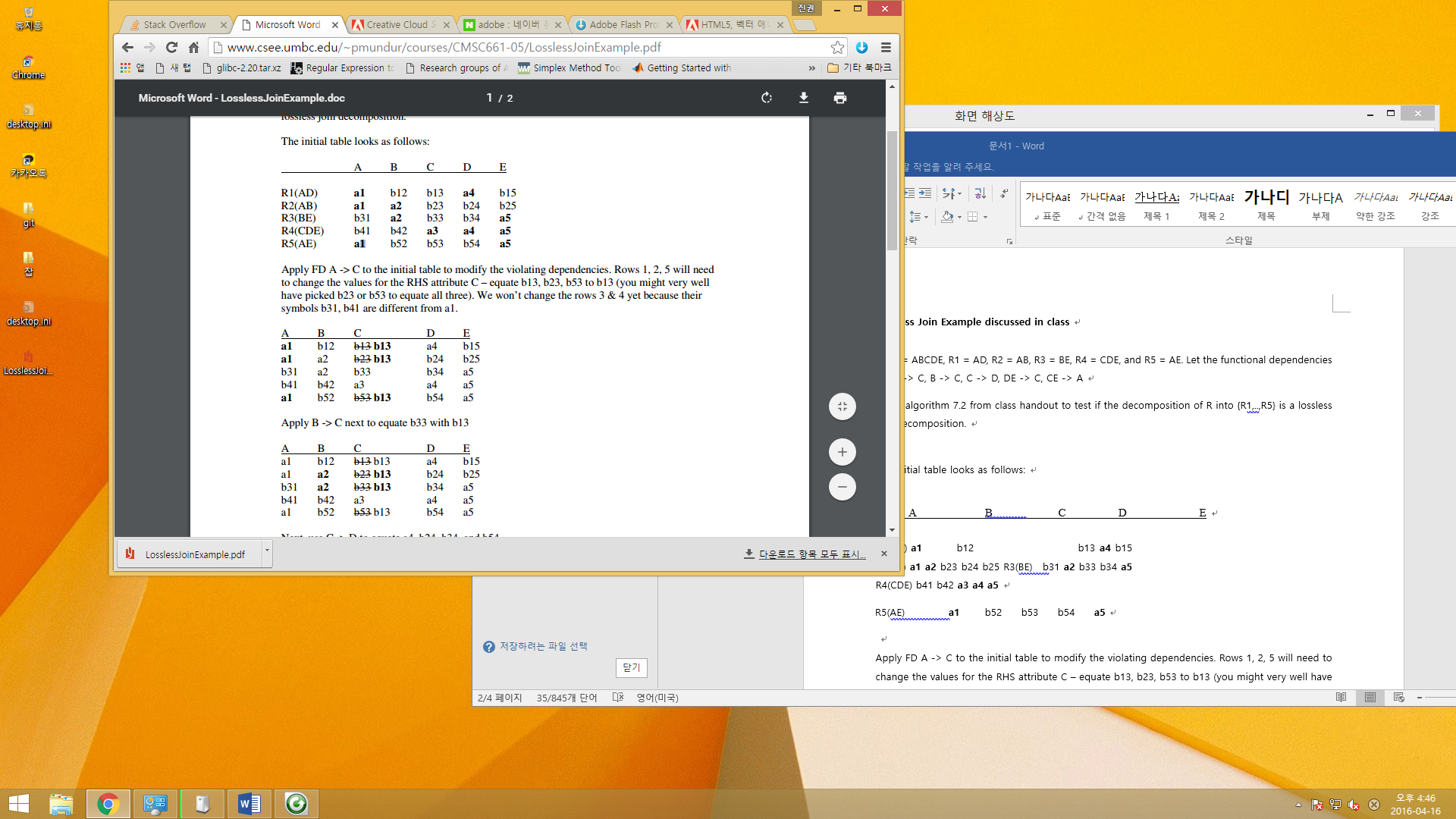
*4. if, after all possible changes have been made to S, a row is made up entirely of a symbols, a(1), a(2), ... , a(n), then the join is lossless. If there is no such row, the join is lossy.*

**Lossless Join Example discussed in class**

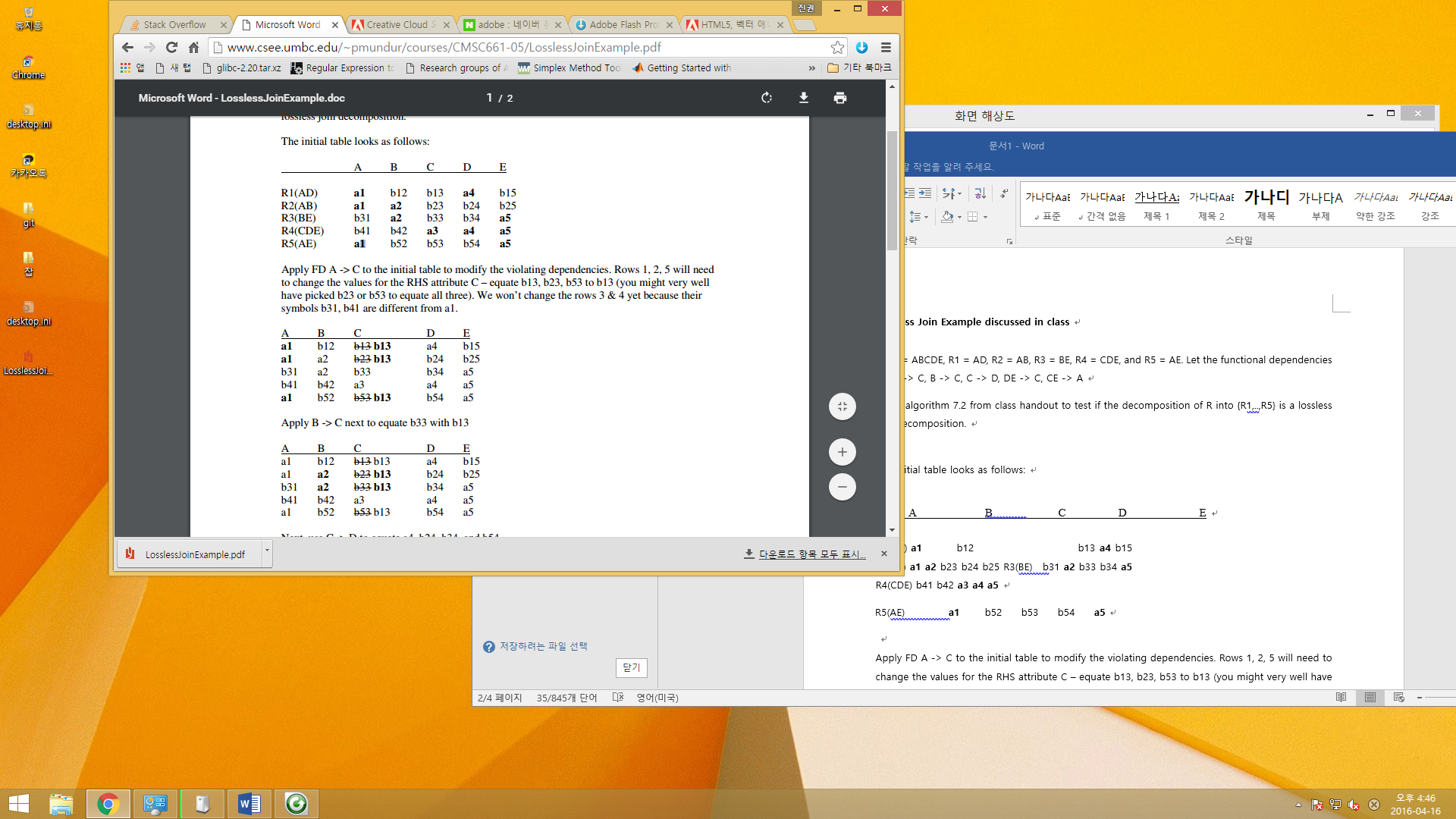
Let R = ABCDE, R1 = AD, R2 = AB, R3 = BE, R4 = CDE, and R5 = AE. Let the functional dependencies be: A -> C, B -> C, C -> D, DE -> C, CE -> A

Apply algorithm 7.2 from class handout to test if the decomposition of R into {R1,..,R5} is a lossless join decomposition.

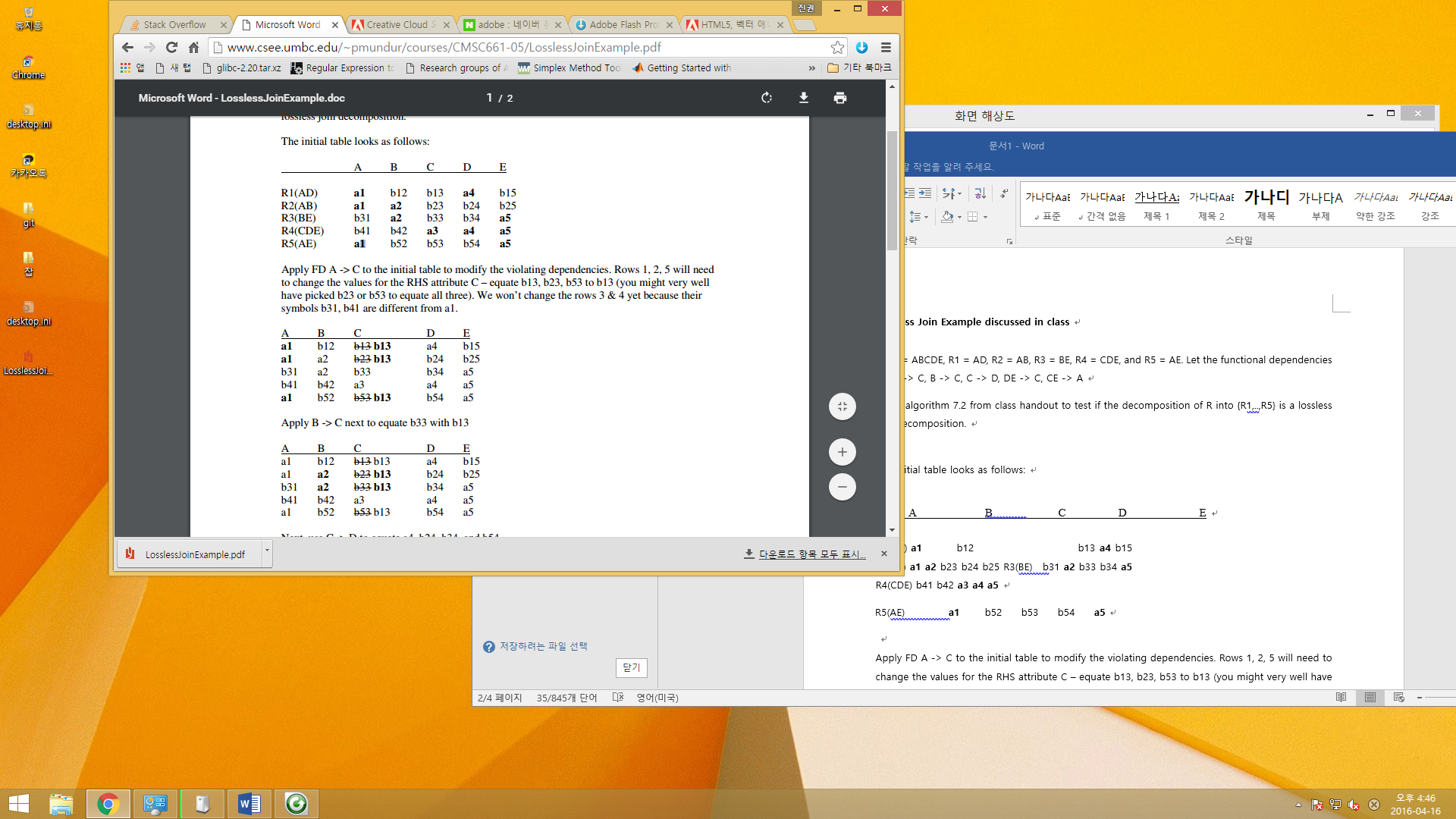
The initial table looks as follows:



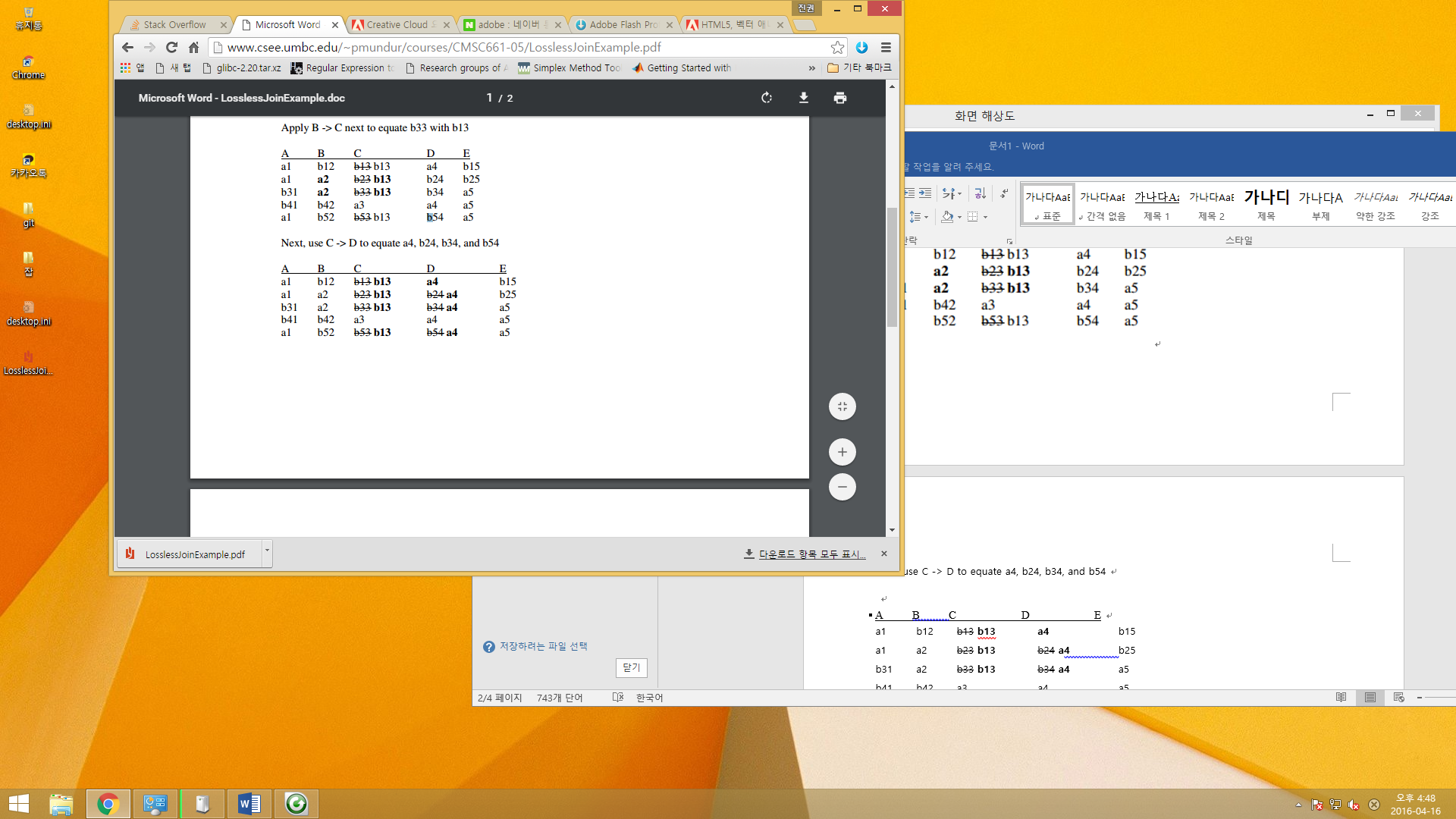
Apply FD A -> C to the initial table to modify the violating dependencies. Rows 1, 2, 5 will need to change the values for the RHS attribute C – equate b13, b23, b53 to b13 (you might very well have picked b23 or b53 to equate all three). We won’t change the rows 3 & 4 yet because their symbols b31, b41 are different from a1.



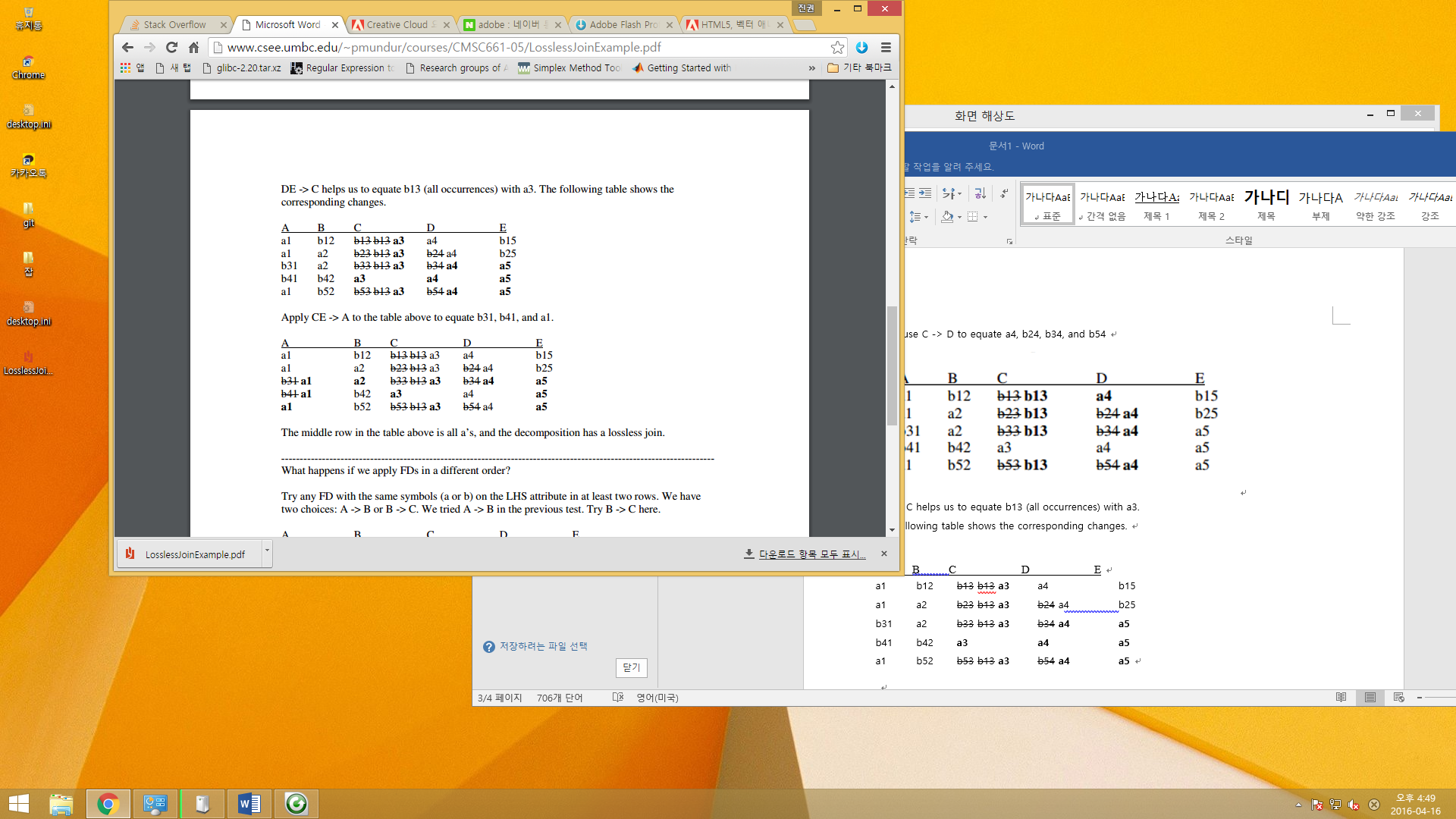
Apply B -> C next to equate b33 with b13



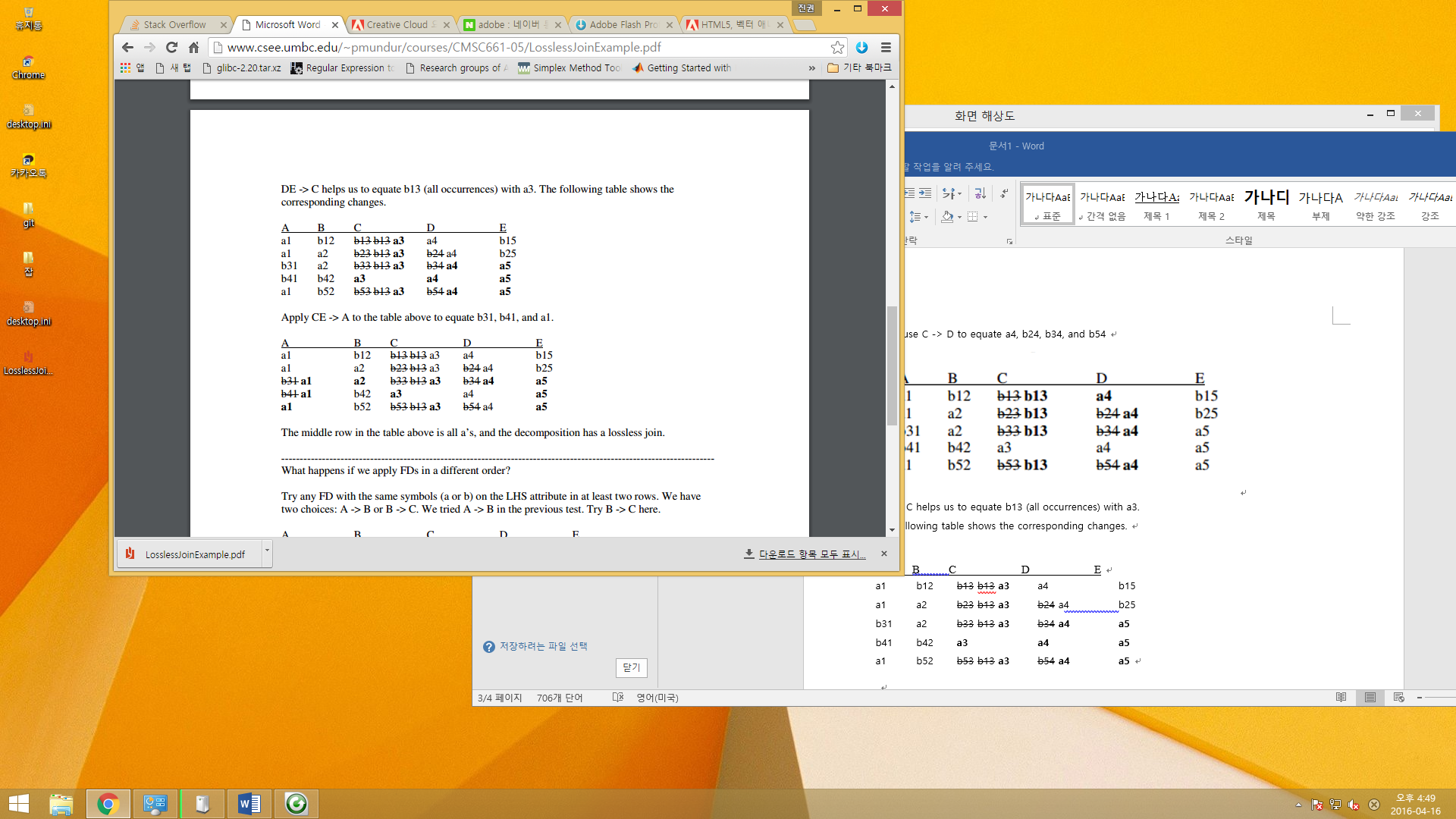
Next, use C -> D to equate a4, b24, b34, and b54



DE -> C helps us to equate b13 (all occurrences) with a3. The following table shows the corresponding changes.



Apply CE -> A to the table above to equate b31, b41, and a1.



The middle row in the table above is all a’s, and the decomposition has a lossless join.