A solution to the question posed in class, which stated
"How do we apply for an inconsistent order of variables?"
The solution to the mentioned question will be listed stepwise along with the an example.

Let us consider the following 2 functions along with the orders as shown:

\[ f = xy + x\bar{y}z \quad \text{Order: } x < y < z \]
\[ g = x\bar{y} + z \quad \text{Order: } z < y < x \]

The ROBDDs of the functions are given below.

![ROBDDs of functions f and g]
Step I:

The order of the final output BDD should be known, or chosen as one of the function's Order. Observe the order of the two functions and place the function with correct order first.

Here, the order of APPLY \( f \) is assumed to be same as that of \( f \) (i.e. \( x < y < z \))

\[
\begin{align*}
f & \rightarrow x < y < z \\
g & \rightarrow z < y < x
\end{align*}
\]

Try overlapping as much as possible, of the second function with the first.

Step II:

Replace all the non-overlapped variables of the second function with some new arbitrary variables.

In the example, for \( g \), we replace \( y \) with \( t \) and \( z \) with \( u \).

\[
\therefore \text{The order of } g \text{ is now } z < t < u \text{, and } g \text{ now takes the form:}
\]

```
     +---+
     |   |
     +---+   +---+
     |   |     |   |
     +---+     +---+
        |       |       |
        +---+       +---+       +---+
        |       |                     |       |
        +---+       +---+                     +---+
```

[Diagram of a BDD with nodes labeled i, j, k, and g]
Step III:

We now merge the nodes of the two functions. Note that the BDD of the functions still remain the same.

\[ x < y < z \]
\[ z < t < u \]
\[ \downarrow \]
\[ x < y < z < t < u \]

Step IV:

We now use the APPLY algorithm on the two trees and obtain a solution for the derived operation.

We proceed to label the nodes and then construct an execution trace. The operation has not been defined in the trace given below.
We now consider the function APPLY as +
Therefore evaluating f + g.

\[ x \]

\[ z \]

\[ y \]

\[ t \]

\[ u \]

\[ \perp \]
Step V: We now put back the original variable names. We then follow the path to the first node that is out of place (order). The values of the variables required to reach the node should be noted. Now we follow the decision paths according to the values recorded, until we reach a '0' or '1'. After evaluating the node structure, we replace the entire segment by its value.

If there are 2 paths (or more), each should be evaluated individually and the appropriate value should be put on that branch.

In the Example,
The new BDD is of order \( x < y < z \) and has follows:

We can now find the ROBDD:

This method can be used when the APPLY function has inconsistent order of variables in the functions involved.