3.4E

4c

No, it does not follow. Sometimes it is the case that there is truth-value assignment that makes P true and one that makes P false, and a truth-value assignment that makes Q true and one that make Q false, but there is no one truth-value assignment such that both P and Q are true on that assignment.

An example: ‘A’ and ‘∼ A’ are both truth-functionally indeterminate, but {‘A’, ‘∼ A’} is not truth-functionally consistent.

3.5E

With many of these questions, you can see what the answer is without constructing the truth-table. You need to construct the truth-table, nevertheless. It’s good for you.

I’ve used the same conventions as the last answers: main connectives singled out by vertical lines around their columns, numbers at the bottom to indicate the order of calculation.

1d

This argument is truth-functionally valid. Here’s the truth-table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>(Y ⇔ A)</th>
<th>Y</th>
<th></th>
<th></th>
<th></th>
<th>W</th>
<th>&amp;</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

You may have the rows in a different order — that’s fine.

As you can see, there are no rows such that all of the premises get assigned T and the conclusion gets assigned F (as there are no rows such that all of the premises get assigned T at all). So the argument is truth-functionally valid.

2c

This argument is truth-functionally valid. Observe:
There is one row that assigns both premises true: the \((F,T)\) row (marked out by horizontal lines).  As you can see, that row assigns \(T\) to the conclusion.  So there are no rows such that all of the premises get assigned \(T\) and the conclusion gets assigned \(F\). So the argument is truth-functionally valid.

2d

Ergh — one of the sentence letters is a ‘T’. Don’t confuse the sentence letter ‘T’ with the truth-value ‘T’ below.

This argument is truth-functionally invalid. Here is a shortened truth-table that shows that.

<table>
<thead>
<tr>
<th>(J)</th>
<th>(M)</th>
<th>(T)</th>
<th>(J \lor (M \supseteq (T \equiv J)))</th>
<th>((M \supseteq J) \land (T \supseteq M))</th>
<th>(T)</th>
<th>(\land \sim M)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(T)</td>
<td>(T)</td>
<td>(T)</td>
<td>(T)</td>
<td>(T)</td>
<td>(T)</td>
<td>(T)</td>
</tr>
</tbody>
</table>

4f

Let \(A = \text{‘The butler murdered Devon’};\) \(B = \text{‘The maid is lying’};\) \(C = \text{‘The gardener murdered Devon’};\) \(D = \text{‘The weapon was a slingshot’}\

The argument:

\[
\frac{(A \supset B) \land (C \supset D) & (B \equiv \sim D) \land (\sim D \supset A)}{A}
\]

This argument is not truth-functionally valid. Check it out:

<table>
<thead>
<tr>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(D)</th>
<th>((A \supset B))</th>
<th>((C \supset D))</th>
<th>((B \equiv \sim D))</th>
<th>((\sim D \supset A))</th>
<th>(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F)</td>
<td>(F)</td>
<td>(T)</td>
<td>(T)</td>
<td>(F)</td>
<td>(T)</td>
<td>(F)</td>
<td>(T)</td>
<td>(F)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5c

No, it does not follow. Here is a counterexample. Let \(P = A \lor B;\) \(Q = A;\) \(R = B;\)

\(\{A \lor B\}\) truth-functionally entails \(A \lor B;\) obviously. \(\{A \lor B\}\) does not truth-functionally entail \(A\) (consider \(A\) false, \(B\) true), and it does not truth-functionally entail \(B\) (consider \(A\) true, \(B\) false).
3.6E

Throughout this section, when I say something like ‘there is no truth-value assignment such that’ blah, what I mean is ‘there is no truth-value assignment such that, on that assignment’ blah.

2b

\( \Gamma \models \neg P \supset Q \) iff there is no truth-value assignment such that every member of \( \Gamma \) is true and \( \neg P \supset Q \) is false. There is no truth-value assignment such that every member of \( \Gamma \) is true and \( \neg P \supset Q \) is false iff there is no truth-value assignment such that every member of \( \Gamma \) is true, \( P \) is true and \( Q \) is false (by the definition of \( \neg \)). There is no truth-value assignment such that every member of \( \Gamma \) is true, \( P \) is true and \( Q \) is false iff there is no truth-value assignment such that every member of \( \Gamma \cup \{ P \} \) is true and \( Q \) is false. There is no truth-value assignment such that every member of \( \Gamma \cup \{ P \} \) is true and \( Q \) is false iff \( \Gamma \cup \{ P \} \models Q \). Therefore, \( \Gamma \models \neg P \supset Q \) iff \( \Gamma \cup \{ P \} \models Q \).

Q.E.D.

3b

Suppose \( \Gamma \models P \) and \( \Gamma \models \neg \neg P \). Then

(1) there is no truth-value assignment such that every member of \( \Gamma \) is true and \( P \) is false, and

(2) there is no truth-value assignment such that every member of \( \Gamma \) is true and \( \neg P \) is false.

By (2) (and the definition of \( \neg \)), there is no truth-value assignment such that every member of \( \Gamma \) is true and \( P \) is true. From this and (1) it follows that there is no truth-value assignment such that every member of \( \Gamma \) is true and \( P \) is true and there is no truth-value assignment such that every member of \( \Gamma \) is true and \( P \) is false. But if there is any truth-value assignment such that every member of \( \Gamma \) is true, it is either such that every member of \( \Gamma \) is true and \( P \) is true or it is such that every member of \( \Gamma \) is true and \( P \) is false. So there is no truth-value assignment such that every member of \( \Gamma \) is true. So \( \Gamma \) is truth-functionally inconsistent.

So if \( \Gamma \models P \) and \( \Gamma \models \neg \neg P \), then \( \Gamma \) is truth-functionally inconsistent.

Q.E.D.

4a

Suppose \( \{ P \} \models Q \), and \( \{ \neg \neg P \} \models R \). Then

(1) there is no truth-value assignment such that \( P \) is true and \( Q \) is false, and

(2) there is no truth-value assignment such that \( \neg P \) is true and \( R \) is false.
By (2) (and the definition of ‘∼’),

(3) there is no truth value assignment such that P is false and R is false.

Now, every truth-value assignment is either such that Q is true or such that Q is false. So, by (1),

(4) if a truth-value assignment is such that P is true, it is such that Q is true.

And every truth-value assignment is either such that R is true or is such that R is false. So, by (3)

(5) if a truth-value assignment is such that P is false, then it is such that R is true.

But every truth-value assignment is either such that P is true or such that P is false. So, by (4) and (5), every truth-value assignment is either such that Q is true or such that R is true. So (by the definition of ‘∨’), every truth-value assignment is such that \( \sim P \lor R \) is true. So \( \sim Q \lor R \) is truth-functionally true.

So if \( \{ P \} \models Q \), and \( \{ \sim P \} \models R \), then \( \sim Q \lor R \) is truth-functionally true. Q.E.D.

4c

Suppose \( \Gamma \models P \) and \( \Gamma' \models Q \). Then

(1) there is no truth-value assignment such that every member of \( \Gamma \) is true and P is false, and

(2) there is no truth value assignment such that every member of \( \Gamma' \) is true and Q is false.

By (1), there is no truth-value assignment such that every member of \( \Gamma \cup \Gamma' \) is true and P is false. And by (2), there is no truth-value assignment such that every member of \( \Gamma \cup \Gamma' \) is true and Q is false. So (by the definition of ‘&’) there is no truth-value assignment such that \( \Gamma \cup \Gamma' \) is true and \( \sim P \land Q \) is false. So \( \Gamma \cup \Gamma' \models \sim P \land Q \).

So, if \( \Gamma \models P \) and \( \Gamma' \models Q \), then \( \Gamma \cup \Gamma' \models \sim P \land Q \).

Q.E.D.