Section 4.1E

1.a.

1	H ∨ G ✓		SM
2	~ G & ~ H ✓		SM
3	~ G		2 &D
4	~ H		2 &D
5	H	G	1 ∨D
	×	×	

Since the truth-tree is closed, the set is truth-functionally inconsistent.

c.

1	H ≡ J				SM
2	~ H ∨ B				SM
3	H		~ H		1 ≡D
4	J		~ J		1 ≡D
5	~ H	J	~ H	J	2 ∨D
6	x	o	o	x	

Since the truth-tree has at least one completed open branch, the set is truth-functionally consistent. The recoverable sets of truth-value assignments are

H	J
T	**T**
F	**F**

e.

1	~ (A ⊃ B) ✓	SM
2	~ (B ⊃ A) ✓	SM
3	A	1 ~ ⊃D
4	~ B	1 ~ ⊃D
5	B	2 ~ ⊃D
6	~ A	2 ~ ⊃D
	×	

Since the truth-tree is closed, the set is truth-functionally inconsistent.

g.

1		B ⊃ (D ⊃ E)		SM
2		D & B ✓		SM
3		D		2 &D
4		B		2 &D
5	~ B		D ⊃ E ✓	1 ⊃D
	×			
6		~ D	E	5 ⊃D
		×	o	

Since the truth-tree has at least one completed open branch, the set is truth-functionally consistent. The recoverable set of truth-value assignments is

B	D	E
T	**T**	**T**

i.

1	H ≡ G ✓		SM
2	~ (H ⊃ G) ✓		SM
3	H		2 ~ ⊃D
4	~ G		2 ~ ⊃D
5	H	~ H	1 ≡D
6	G	~ G	1 ≡D
	×	×	

Since the truth-tree is closed, the set is truth-functionally inconsistent.

k.

1	L≡ (J & K)] ✓			SM
2	~ J			SM
3	~ L ⊃ L ✓			SM
4	L	~ L		1 ≡D
5	J & K ✓	~ (J & K)		1 ≡D
6	J			5 &D
7	K			5 &D
	×			
8		~ ~ L	L	3 ⊃D
9		L	×	8 ~~D
		×		

Since the truth-tree is closed, the set is truth-functionally inconsistent.

m. 1		~ [(A ≡ B) ≡ A)] ✓			SM
2	A ≡ B ✓		~ (A ≡ B) ✓		1 ~ ≡ D
3	~ A		A		1 ~ ≡ D
4	A	~ A			2 ≡D
5	B	~ B			2 ≡D
	×	o			
6			A	~ A	2 ~ ≡D
7			~ B	B	2 ~ ≡D
			o	×	

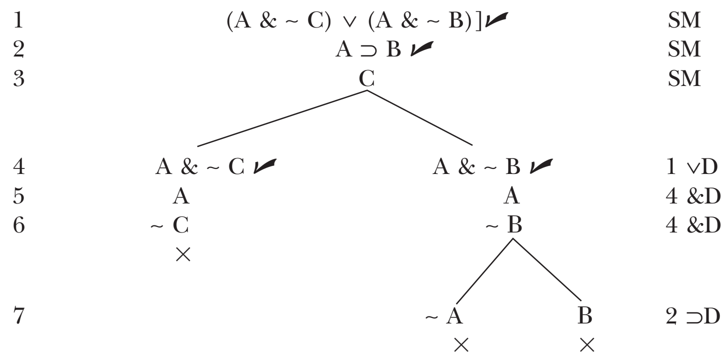
Since the truth-tree has at least one completed open branch, the set is truth-functionally consistent. The recoverable sets of truth-value assignments are

A	B
T	**F**
F	**F**

o. 1	~ [(A ⊃ ~ B) ⊃ (B ⊃ A)] ✓		SM
2	~ (~ A ⊃ ~ B) ✓		SM
3	A ⊃ ~ B ✓		1 ~ ⊃ I
4	~ (B ⊃ A)] ✓		1 ~ ⊃ I
5	B		4 ~ ⊃ I
6	~ A		4 ~ ⊃ I
7	~ A		2 ~ ⊃ I
8	~ ~ B		2 ~ ⊃ I
9	~ A	~ B	3 ⊃D
	B	×	8 ~ ~D
	o		

Since the truth-tree has at least one completed open branch, the set is truth-functionally consistent. The recoverable fragment is

A	B
F	**T**

q.


Since the truth-tree is closed, the set is truth-functionally inconsistent.

2.a. True. Truth-trees test for consistency. A completed open branch shows that the set is consistent because it yields at least one truth-value assignment on which all the members of the set being tested are true. An open branch on a completed truth-tree is a completed open branch.

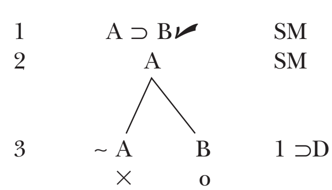
c. True. If a tree has a completed open branch, then we can recover from that branch a set of truth-value assignments on which every member of the set is true. And a set is, by definition, consistent if and only if there is at least one truth-value assignment on which all its members are true.

e. True. If all the branches are closed, there is no truth-value assignment on which all the members of the set being tested are true, and if there is no such assignment, that set is truth-functionally inconsistent.

g. False. The number of branches on a completed tree and the number of distinct atomic components of the members of the set being tested are not related.

i. False. Closed branches represent unsuccessful attempts to find truth-value assignments on which all the members of the set being tested are true. No sets of truth-value assignments are recoverable from them; hence they do not yield assignments on which all the members of the set being tested are false.

k. False. The truth-tree for {A ⊃ B, A} has a closed branch.



Section 4.2E

1. a.

1	M & ~ M✓	SM
2	M	1 &D
3	~ M	1 &D
	×	

Since the truth-tree for the given sentence is closed, that sentence is truth-functionally false.

c.

1	~ M ∨ ~ M✓		SM
2	~ M	~ M	1 ∨D
	o	o	

1	~ (~ M ∨ ~ M)✓	SM
2	~ ~ M✓	1 ~ ∨D
3	~ ~ M✓	1 ~ ∨D
4	M	2 ~ ~ D
5	M	3 ~ ~ D
	o	

Neither the truth-tree for the given sentence nor the truth-tree for the negation of that sentence is closed, therefore the given sentence is truth-functionally indeterminate.

e.

1	(C ⊃ R) & [(C ⊃ ~ R) & ~ (~ C ∨ R)]✓		SM
2	C ⊃ R✓		1 &D
3	(C ⊃ ~ R) & ~ (~ C ∨ R)✓		1 &D
4	C ⊃ ~ R		3 &D
5	~ (~ C ∨ R)✓		3 &D
6	~ ~ C✓		5 ~ ∨D
7	~ R		5 ~ ∨D
8	C		6 ~ ~ D
9	~ C	R	2 ⊃D
	×	×	

Since the truth-tree is closed, the sentence we are testing is truth-functionally false.

g. 1	(~ A ≡ ~ Z) & (A & ~ Z) ✓		SM
2	~ A ≡ ~ Z ✓		1 &D
3	A & ~ Z ✓		1 &D
4	A		3 &D
5	~ Z		3 &D
6	~ A	~ ~ A	2 ≡D
7	~ Z	~ ~ Z ✓	2 ≡D
	×		
8		Z	7 ~ ~ D
		×	

Since the truth-tree is closed, the sentence we are testing is truth-functionally false.

i. 1	(A ∨ B) & ~ (A ∨ B) ✓		SM
2	A ∨ B ✓		1 &D
3	~ (A ∨ B) ✓		1 &D
4	~ A		3 ~ ∨D
5	~ B		3 ~ ∨D
6	A	B	2 ∨D
	×	×	

The tree is closed, so the sentence is truth-functionally false.

k. 1	(A ∨ B) ≡ ~ (A ∨ B) ✓				SM
2	A ∨ B ✓		~ (A ∨ B) ✓		1 ≡D
3	~ (A ∨ B) ✓		~ ~ (A ∨ B) ✓		1 ≡D
4	~ A				3 ~ ∨D
5	~ B				3 ~ ∨D
6	A	B			2 ∨D
7	×	×	~ A		2 ~ ∨D
8			~ B		2 ~ ∨D
9			A ∨ B ✓		3 ~ ~ D
10			A	B	9 ∨D
			×	×	

The tree is closed, so the sentence is truth-functionally false.

```
m.  1            ~ (D ∨ F) ≡ (~ D ∨ ~ F)                 SM
                      /              \
    2        ~ (D ∨ F)✓         ~ ~ (D ∨ F)✓             1 ≡D
    3        ~ D ∨ ~ F✓      ~ (~ D ∨ ~ F)✓              1 ≡D
    4            |                 D ∨ F✓                2 ~ ~ D
    5            |                 ~ ~ D✓                3 ~ ∨D
    6            |                 ~ ~ F✓                3 ~ ∨D
    7            |                   D                   5 ~ ~ D
    8            |                   F                   6 ~ ~ D
                 |                  / \
    9            |                 D   F                 4 ∨D
   10           ~ D                o   o                 2 ~ ∨D
   11           ~ F                                      2 ~ ∨D
               /   \
   12        ~ D   ~ F                                   3 ∨ D
              o     o
```

```
    1              ~ (~ (D ∨ F) ≡ (~ D ∨ ~ F))✓                     SM
                       /                    \
    2          ~ (D ∨ F)✓              ~ ~ (D ∨ F)✓                 1 ~ ≡D
    3        ~ (~ D ∨ ~ F)✓             ~ D ∨ ~ F✓                  1 ~ ≡D
    4              |                      D ∨ F✓                    2 ~ ~ D
                   |                   /          \
    5              |                 ~ D          ~ F               3 ∨D
                   |                /   \        /   \
    6              |               D     F      D     F             4 ∨D
    7             ~ D              ×     o      o     ×             2 ~ ∨D
    8             ~ F                                               2 ~ ∨D
    9            ~ ~ D✓                                             3 ~ ∨D
   10            ~ ~ F                                              3 ~ ∨D
   11              D                                                9 ~ ~ D
                   ×
```

Neither the tree for the sentence nor the tree for its negation is closed. Therefore the sentence is truth-functionally indeterminate.

2.a. 1	$\sim [(B \supset L) \vee (L \supset B)]$ ✓	SM
2	$\sim (B \supset L)$ ✓	$1 \sim \vee D$
3	$\sim (L \supset B)$ ✓	$1 \sim \vee D$
4	B	$2 \sim \supset D$
5	$\sim L$	$2 \sim \supset D$
6	L	$3 \sim \supset D$
7	$\sim B$	$3 \sim \supset D$
	×	

Since the truth-tree for the negation of the given sentence is closed, the given sentence is truth-functionally true.

c. 1	$\sim [(A \equiv K) \supset (A \vee K)]$ ✓		SM
2	$A \equiv K$ ✓		$1 \sim \supset D$
3	$\sim (A \vee K)$ ✓		$1 \sim \supset D$
4	$\sim A$		$3 \sim \vee D$
5	$\sim K$		$3 \sim \vee D$
6	A	$\sim A$	$2 \equiv D$
7	K	$\sim K$	$2 \equiv D$
	×	o	

Since the truth-tree for the negation of the given sentence is not closed, the given sentence is not truth-functionally true. The recoverable set of truth-value assignments is

A	K
F	**F**

e. 1	$\sim [[(J \supset Z) \ \& \sim Z] \supset \sim J]$ ✓		SM
2	$(J \supset Z) \ \& \sim Z$ ✓		$1 \sim \supset D$
3	$\sim \sim J$ ✓		$1 \sim \supset D$
4	J		$3 \sim \sim D$
5	$J \supset Z$ ✓		2 &D
6	$\sim Z$		2 &D
7	$\sim J$	Z	$5 \supset D$
	×	×	

Since the truth-tree for the negation of the given sentence is closed, the given sentence is truth-functionally true.

g.

1	~ [(B ⊃ (M ⊃ H)) ≡ [(B ⊃ M) ⊃ (B ⊃ H)]]✔				SM
2	B ⊃ (M ⊃ H)✔		~(B ⊃ (M ⊃ H))✔		1 ~ ≡D
3	~ [{B ⊃ M) ⊃ (B ⊃ H)]✔		(B ⊃ M) ⊃ (B ⊃ H)✔		1 ~ ≡D
4	B ⊃ M✔				3 ~ ⊃D
5	~ (B ⊃ H)✔				3 ~ ⊃D
6	B				5 ~ ⊃D
7	~ H				5 ~ ⊃D
8	~ B ×	M			4 ⊃D
9	~ B ×	M ⊃ H✔			2 ⊃D
10	~ M ×	H ×			9 ⊃D
11			B		2 ~ ⊃D
12			~ (M ⊃ H)✔		2 ~ ⊃D
13			M		12 ~ ⊃D
14			~ H		12 ~ ⊃D
15			~ (B ⊃ M)✔	B ⊃ H✔	3 ⊃D
16			B		15 ~ ⊃D
17			~ M ×		15 ~ ⊃D
18			~ B ×	H ×	15 ⊃D

Since the truth-tree for the negation of the given sentence is closed, the given sentence is truth-functionally true.

i.

1	~ ([(A ⊃ B) ⊃ A] ⊃ A)✔		SM
2	(A ⊃ B) ⊃ A✔		1 ~ ⊃D
3	~ A		1 ~ ⊃D
4	~ (A ⊃ B)	A ×	3 ⊃D
5	A		4 ~ ⊃D
6	~ B ×		4 ~ ⊃D

Since the truth-tree for the negation of the given sentence is closed, the given sentence is truth-functionally true.

k.

1	~ (((A & B) ⊃ C) ≡ ((A ⊃ ~ B) ∨ C))✓				SM
2	(A & B) ⊃ C✓		~ ((A & B) ⊃ C)✓		1 ~ ≡D
3	~ ((A ⊃ ~ B) ∨ C)✓		(A ⊃ ~ B) ∨ C✓		1 ~ ≡D
4	~ (A ⊃ ~ B)✓				3 ~ ∨D
5	~ C				3 ~ ∨D
6	A				4 ~ ⊃D
7	~ ~ B✓				4 ~ ⊃D
8	B				7 ~ ~ D
9	~ (A & B)✓	C ×			2 ⊃D
10	~ A × / ~ B ×				9 ~ &D
11			A & B✓		2 ~ ⊃D
12			~ C		2 ~ ⊃D
13			A		11 &D
14			B		11 &D
15			A ⊃ ~ B✓	C ×	3 ∨D
16			~ A × / ~ B ×		15 ⊃D

The tree for the negation of the sentence is closed. Therefore the sentence is truth-functionally true.

m.

1	~ ((A ⊃ (B & C)) ⊃ (A ⊃ (B ⊃ C)))✓		SM
2	A ⊃ (B & C)✓		1 ~ ⊃D
3	~ (A ⊃ (B ⊃ C))✓		1 ~ ⊃D
4	A		3 ~ ⊃D
5	~ (B ⊃C)✓		3 ~ ⊃D
6	B		5 ~ ⊃D
7	~ C		5 ~ ⊃D
8	~ A ×	B & C✓	2 ⊃D
9		B	8 &D
10		C ×	8 &D

The tree for the negation of the sentence is closed. Therefore the sentence is truth-functionally true.

o. 1	$\sim (((A \& B) \supset C) \equiv (A \supset (B \supset C)))$ ✓			SM
2	$(A \& B) \supset C$ ✓		$\sim ((A \& B) \supset C)$ ✓	$1 \sim \equiv D$
3	$\sim (A \supset (B \supset C))$ ✓		$A \supset (B \supset C)$ ✓	$1 \sim \equiv D$
4	A			$3 \sim \supset D$
5	$\sim (B \supset C)$ ✓			$3 \sim \supset D$
6	B			$5 \sim \supset D$
7	$\sim C$			$5 \sim \supset D$
8	$\sim (A \& B)$ ✓	C ×		$2 \supset D$
9	$\sim A$ × $\sim B$ ×			$8 \sim \& D$
10			$A \& B$ ✓	$2 \sim \supset D$
11			$\sim C$	$2 \sim \supset D$
12			A	$10 \ \& D$
13			B	$10 \ \& D$
14			$\sim A$ × $B \supset C$ ✓	$3 \supset D$
15			$\sim B$ × C ×	$14 \supset D$

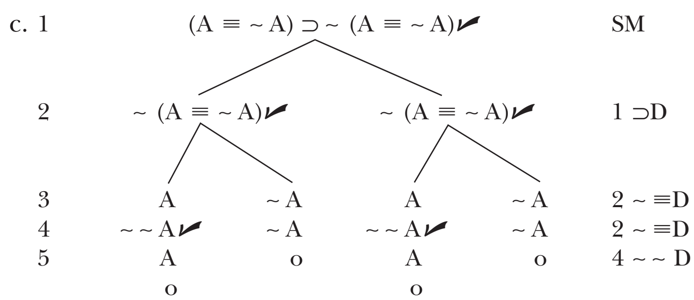
The tree for the negation of the sentence is closed. Therefore the sentence is truth-functionally true.

3.a. 1	$\sim (\sim A \supset A)$ ✓	SM
2	$\sim A$	$1 \sim \supset D$
3	$\sim A$	$1 \sim \supset D$
	o	

The tree for the sentence does not close. Therefore the sentence is not truth-functionally false. The recoverable set of truth-value assignments is

A
F

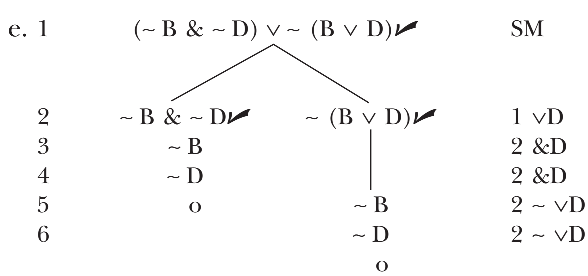
Since not all sets of truth-value assignments are recoverable, the sentence is not truth-functionally true. Therefore it is truth-functionally indeterminate.



The tree for the sentence does not close. Therefore the sentence is not truth-functionally false. The recoverable sets of truth-value assignments are

A
T
F

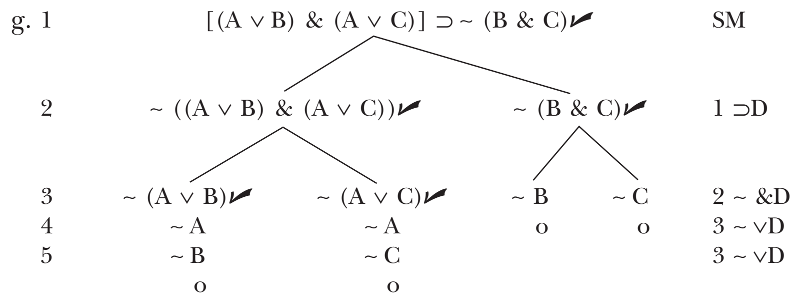
Since all sets of truth-value assignments are recoverable, the sentence is truth-functionally true.



The tree for the sentence does not close. Therefore the sentence is not truth-functionally false. The recoverable set of truth-value assignments is

B	D
F	**F**

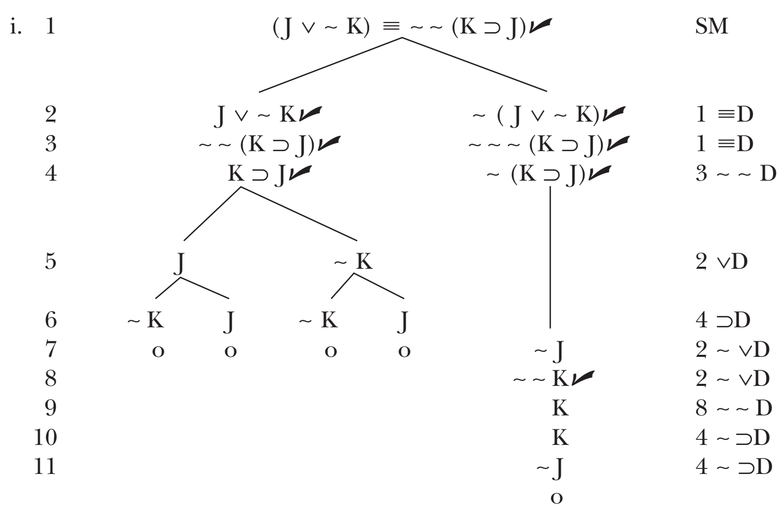
Since not all sets of truth-value assignments are recoverable, the sentence is not truth-functionally true. Therefore it is truth-functionally indeterminate.



The tree for the sentence does not close. Therefore the sentence is not truth-functionally false. The recoverable sets of truth-value assignments are

A	B	C
F	**F**	**T**
F	**F**	**F**
F	**T**	**F**
T	**F**	**T**
T	**F**	**F**
T	**T**	**F**

Since not all truth-value assignments are recoverable, the sentence is not truth-functionally true. Therefore it is truth-functionally indeterminate.



The tree for the sentence does not close. Therefore the sentence is not truth-functionally false. The recoverable sets of truth-value assignments are

J	K
T	**F**
T	**T**
F	**T**
F	**F**

Since all truth-value assignments are recoverable, the sentence is truth-functionally true.

4.a. False. A tree for a truth-functionally true sentence can have some open and some closed branches. '(H ∨ ~ H) ∨ (~ H & H)' is clearly truth-functionally true, inasmuch as its left disjunct is truth-functionally true. Yet the tree for this sentence has two open branches and one closed branch.

```
1         (H ∨ ~ H) ∨ (~ H & H)✓        SM
                  /        \
2       H ∨ ~ H✓          ~ H & H✓      1 ∨D
          /  \
3        /    \             ~ H         2 &D
4       /      \              H         2 &D
       /        \             ×
5     H         ~ H                     2 ∨D
      o          o
```

c. False. Many truth-functionally indeterminate sentences have completed trees all of whose branches are open. A simple example is

```
1     H ∨ G✓      SM
       /  \
2     H    G      1 ∨D
      o    o
```

e. False. Some such unit sets have open trees; for example, **P** ∨ **Q** does, but not all such unit sets have open trees. For example, **P** & **Q** has a closed tree if **P** is 'H & G' and **Q** is '~ H & K'.

1	(H & G) & (~ H & K)✓	SM
2	H & G✓	1 &D
3	~ H & K✓	1 &D
4	H	2 &D
5	G	2 &D
6	~ H	3 &D
7	K	3 &D
	×	

g. The claim is false. If **P** and **Q** are both truth-functionally true, then **P** & **Q**, **P** ∨ **Q**, **P** ⊃ **Q**, and **P** ≡ **Q** are also truth-functionally true. Therefore the unit set of each is truth-functionally consistent and will not have a closed truth-tree. But each may still have a tree with one or more closed branches. For example, if **P** is '(A ∨ ~ A) ∨ (B & ~ B)' then **P** & **Q**, **P** ∨ **Q**, and **P** ≡ **Q** will each have at least one closed branch—the one resulting from the decomposition of 'B & ~ B'. And if **P** is 'A ∨ ~ A' and **Q** is 'B ∨ ~ B', then the tree for **P** ⊃ **Q** will have a closed branch, the one resulting from the occurrence of '~ (A ∨ ~ A)' on line 2 of the tree for this sentence.

i. The claim is false. Given that both **P** and **Q** are truth-functionally false, **P** & **Q** and **P** ∨ **Q** will also be truth-functionally false, and hence will have closed truth-trees. However, **P** ⊃ **Q** and **P** ≡ **Q** will both be truth-functionally true. (The only way **P** ⊃ **Q** could fail to be truth-functionally true would be for there to be a truth-value assignment on which **P** is true and **Q** is false, but there is no truth-value assignment on which **P** is true since **P** is truth-functionally false. The only way **P** ≡ **Q** could fail to be truth-functionally true would be for there to be a truth-value assignment on which **P** and **Q** have different truth-values. But then there would have to be an assignment on which one of **P** and **Q** is true, but there can be no such assignment since both **P** and **Q** are truth-functionally false.) And sentences that are truth-functionally true have completed truth-trees that are open, not closed.

k. The claim is false. If **P** is, as stated, truth-functionally true and **Q** is truth-functionally false, then **P** & **Q**, **P** ⊃ **Q**, and **P** ≡ **Q** will all be truth-functionally false. **P** & **Q** so because there will be no truth-value assignment on which **P** and **Q** are both true (because **Q** is truth-functionally false. Hence **P** & **Q** will have a closed truth-tree (one on which every branch is closed).

Similarly, **P** ⊃ **Q** will be false on every truth-value assignment because **P** will be true and **Q** false on every assignment. So the tree for **P** ⊃ **Q** will also be closed. **P** ≡ **Q** will be truth-functionally false because on every truth-value assignment **P** will be true and **Q** false, so there will be no assignment on which **P** and **Q** have the same truth-value, that is, no assignment on which **P** ≡ **Q** is true. So the tree for **P** ≡ **Q** will be closed. However, **P** ∨ **Q** will be truth-functionally true, because **P** is truth-functionally true. Line 2 of the tree will contain **P** on the left branch and **Q** on the right. Because **P** is truth-functionally true, subsequent work on the left branch will yield at least one (in fact at least two) completed open branch (see answer to exercise h). The right branch, that which has **Q** at the top, will become a closed branch because **Q** is truth-functionally false.

5.a.

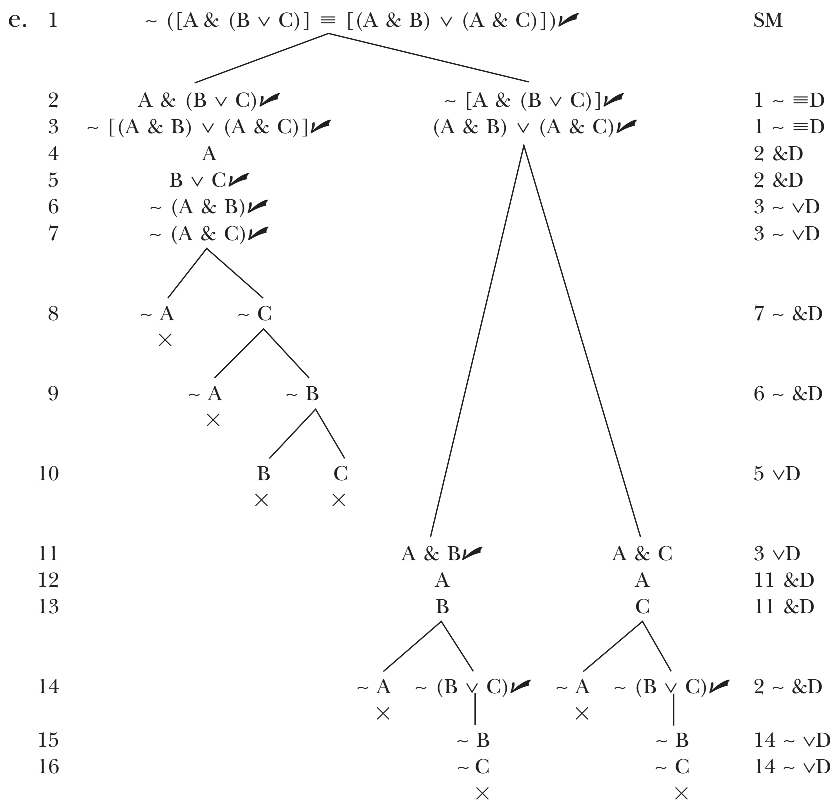
Line					
1	~ [~ (Z ∨ K) ≡ (~ Z & ~ K)]✓				SM
2	~ (Z ∨ K)✓		~ ~ (Z ∨ K)✓		1 ~ ≡D
3	~ (~ Z & ~ K)✓		~ Z & ~ K✓		1 ~ ≡D
4	~ Z				2 ~ ∨D
5	~ K				2 ~ ∨D
6	~ ~ Z✓	~ ~ K✓			3 ~ &D
7	Z	K			6 ~ ~ D
	×	×			
8			Z ∨ K✓		2 ~ ~ D
9			~ Z		3 &D
10			~ K		3 &D
11			Z	K	8 ∨D
			×	×	

Our truth-tree for the negation of the biconditional of the sentences we are testing, '~ (Z ∨ K)' and '~ Z & ~ K', is closed. Therefore that negation is truth-functionally false, the biconditional it is a negation of is truth-functional true, and the sentences we are testing are truth-functionally equivalent.

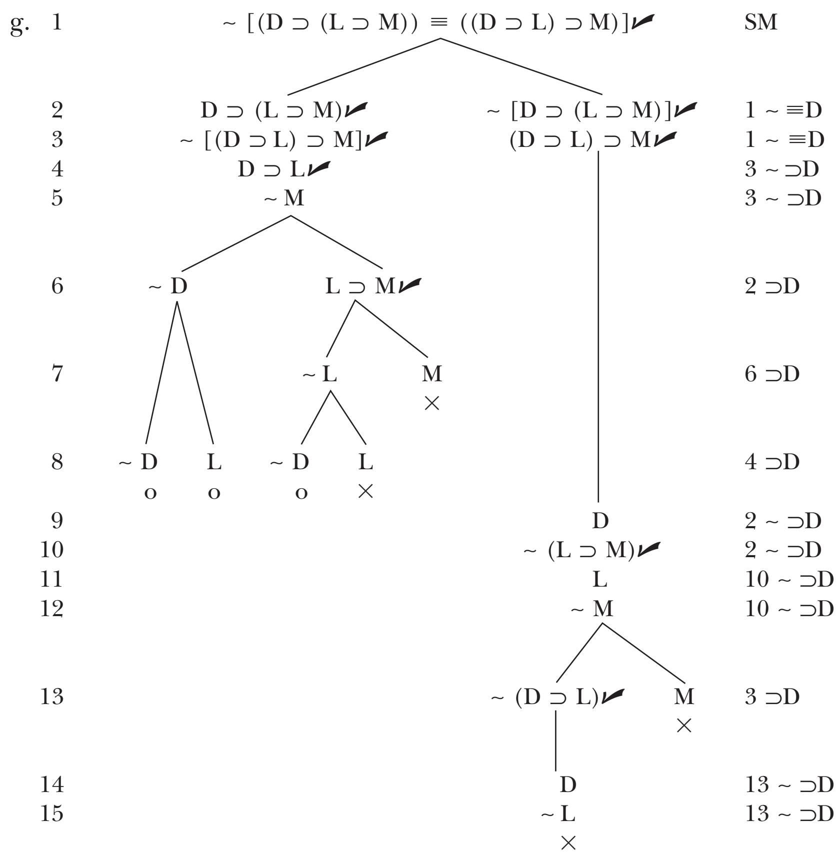
c. 1	~ [[(B & C) ⊃ R] ≡ [(B ⊃ R) & (C ⊃ R)]] ✓						SM
2	(B & C) ⊃ R ✓				~ [B & C) ⊃ R] ✓		1 ~ ≡D
3	~ [(B ⊃ R) & (C ⊃ R)]				(B ⊃ R) & (C ⊃ R) ✓		1 ~ ≡D
4	~ (B ⊃ R) ✓		~ (C ⊃ R) ✓				3 ~ &D
5	B		C				4 ~ ⊃D
6	~ R		~ R				4 ~ ⊃D
7	~ (B & C) ✓	R ×	~ (B & C) ✓	R ×			2 ⊃D
8	~ B × / ~ C o		~ B o / ~ C ×				7 ~ &D
9					B & C ✓		2 ~ ⊃D
10					~ R		2 ~ ⊃D
11					B		9 &D
12					C		9 &D
13					B ⊃ R ✓		3 &D
14					C ⊃ R		3 &D
15					~ B ×	R ×	13 ⊃D

Since our truth-tree for the negation of the biconditional of the sentences we are testing is open, those sentences are not truth-functionally equivalent. The recoverable sets of truth-value assignments are

B	C	R
T	**F**	**F**
F	**T**	**F**



Since our truth-tree for the negation of the biconditional of the sentences we are testing is closed, those sentences are truth-functionally equivalent.



Since our truth-tree for the negation of the biconditional of the sentences we are testing is open, those sentences are not truth-functionally equivalent. The recoverable sets of truth-value assignments are

D	L	M
F	**T**	**F**
F	**F**	**F**

i.

1	~ [(A ⊃ A) ≡ (B ⊃ B)] ✓		SM
2	A ⊃ A	~ (A ⊃ A) ✓	1 ~ ≡D
3	~ (B ⊃ B) ✓	B ⊃ B	1 ~ ≡D
4	B		3 ~ ⊃D
5	~ B		3 ~ ⊃D
6	×	A	2 ~ ⊃D
7		~ A	2 ~ ⊃D
		×	

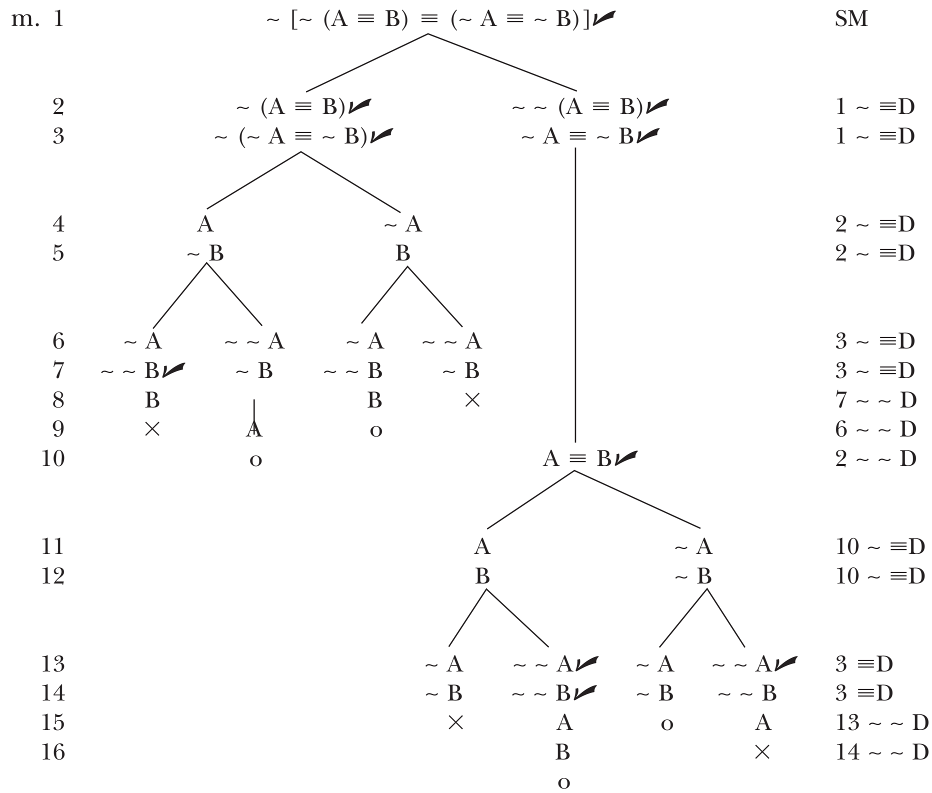
Since the truth-tree is closed, the sentences being tested are truth-functionally equivalent.

k.

1	~ [(A & ~ B) ≡ (~ A ∨ B)] ✓					SM
2	A & ~ B ✓	~ (A & ~ B) ✓				1 ~ ≡D
3	~ (~ A ∨ B) ✓	~ A ∨ B ✓				1 ~ ≡D
4	A					2 &D
5	~ B					2 &D
6	~ ~ A ✓					3 ~ ∨D
7	~ B					3 ~ ∨D
8	A					6 ~ ~ D
9	o	~ A		~ ~ B		2 ~ &D
10				B		9 ~ ~ D
11		~ A	B	~ A	B	3 ∨D
		o	o	o	o	

Since the truth-tree is not closed, the sentences being tested are not truth-functionally equivalent. The recoverable sets of truth-value assignments are

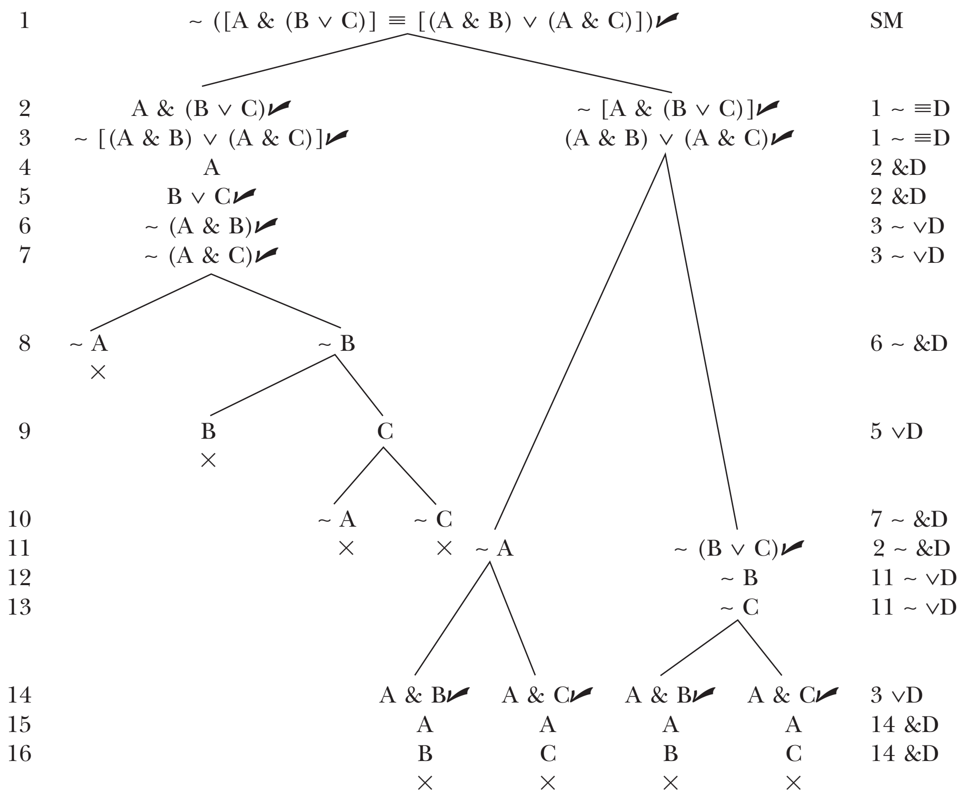
A	B
T	**F**
F	**F**
F	**T**
T	**T**



Since the truth-tree has at least one completed open branch, the sentences being tested are not truth-functionally equivalent. The recoverable sets of truth-value assignments are

A	B
T	**T**
F	**F**
T	**F**
F	**F**

o.



Since the truth-tree is closed, the sentences being tested are truth-functionally equivalent.

6.a. True. If **P** and **Q** are truth-functionally equivalent, their biconditional is truth-functionally true. And all truth-functionally true sentences have completed open trees.

c. False. The tree for the set {**P**, **Q**} may close, for **P** and **Q** may both be truth-functionally false. Remember that all truth-functionally false sentences are truth-functionally equivalent and a set composed of one or more truth-functionally false sentences has a closed tree.

7.a. 1	A ⊃ (B & C)✔		SM
2	C ≡ B		SM
3	~ C		SM
4	~ ~ A✔		SM
5	A		4 ~ ~ D
6	~ A	B & C✔	1 ⊃D
7	×	B	6 &D
8		C	6 &D
		×	

Our tree is closed, so the set {A ⊃ (B & C), C ≡ B, ~ C} does truth-functionally entail '~ A'.

c. 1	~ (A ≡ B)✔		SM
2	~ A		SM
3	~ B		SM
4	~ (C & ~ C)		SM
5	A	~ A	1 ~ ≡D
6	~ B	B	1 ~ ≡D
	×	×	

Our tree is closed, so the set {~ (A ≡ B), ~ A, ~ B} does truth-functionally entail 'C & ~ C'.

e. 1	~ ~ F ⊃ ~ ~ G✔				SM
2	~ G ⊃ ~ F✔				SM
3	~ (G ⊃ F)✔				SM
4	G				3 ~ ⊃D
5	~ F				3 ~ ⊃D
6	~ ~ ~ F✔		~ ~ G✔		1 ⊃D
7	~ F		G		6 ~ ~ D
8	~ ~ G✔	~ F	~ ~ G✔	~ F	2 ⊃D
9	G	o	G	o	8 ~ ~ D
	o		o		

Our truth-tree is open, so the set {~ ~ F ⊃ ~ ~ G, ~ G ⊃ ~ F} does not truth-functionally entail ‘G ⊃ F’. The recoverable set of truth-value assignments is

F	G
F	**T**

g.

1	[(C ∨ D) & H] ⊃ A✓		SM
2	D		SM
3	~ (H ⊃ A)✓		SM
4	H		3 ~ ⊃D
5	~ A		3 ~ ⊃D
6	~ [(C ∨ D) & H]✓	A ×	1 ⊃D
7	~ (C ∨ D)✓ ~ H ×		6 ~ &D
8	~ C		7 ~ ∨D
9	~ D ×		7 ~ ∨D

Our truth-tree is closed, so the given set does truth-functionally entail ‘H ⊃ A’.

i.

1		(J ∨ M) ⊃ ~ (J & M)✓			SM
2		M ≡ (M ⊃ J)✓			SM
3		~ (M ⊃ J)✓			SM
4		M			3 ~ ⊃D
5		~ J			3 ~ ⊃D
6	~ (J ∨ M)✓	~ (J & M)✓			1 ⊃D
7	~ J				6 ~ ∨D
8	~ M				6 ~ ∨D
9	×	M		~ M	2 ≡D
10		M ⊃ J✓		~ (M ⊃ J) ×	2 ≡D
11		~ J	~ M ×		6 ~ &D
12		~ M J × ×			10 ⊃D

The tree is closed, so the set {(J ∨ M) ⊃ ~ (J & M), M ≡ (M ⊃ J)} does truth-functionally entail ‘M ⊃ J’.

```
k.  1            ~ (~ (A ≡ B) ⊃ (~ A ≡ ~ B))✔           SM
    2                   ~ (A ≡ B)✔                       1 ~ ⊃D
    3                 ~ (~ A ≡ ~ B)✔                     1 ~ ⊃D
                    /               \
    4              A                 ~ A                 2 ~ ≡D
    5             ~ B                  B                 2 ~ ≡D
                /     \             /     \
    6        ~ A     ~ ~ A✔       ~ A     ~ ~ A✔         3 ~ ≡D
    7       ~ ~ B     ~ B        ~ ~ B✔    ~ B           3 ~ ≡D
    8         ×        A           |        A            6 ~ ~ D
    9                  o           B        ×            7 ~ ~ D
                                   o
```

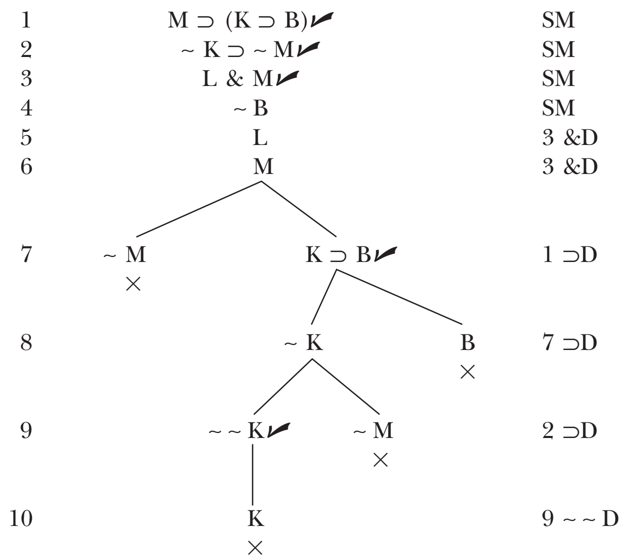
Our truth-tree is open, so the empty set does not truth-functionally entail '~ (A ≡ B) ⊃ (~ A ≡ ~ B)'. The recoverable sets of truth-value assignments are

A	B
T	**F**
F	**T**

```
m.  1     ~ (((A ⊃ B) ⊃ (C ⊃ D)) ⊃ (C ⊃ (B ⊃ D)))✔       SM
    2             (A ⊃ B) ⊃ (C ⊃ D)✔                     1 ~ ⊃D
    3              ~ (C ⊃ (B ⊃ D))✔                      1 ~ ⊃D
    4                      C                             3 ~ ⊃D
    5                 ~ (B ⊃ D)✔                         3 ~ ⊃D
    6                      B                             5 ~ ⊃D
    7                     ~ D                            5 ~ ⊃D
                     /           \
    8         ~ (A ⊃ B)✔        C ⊃ D✔                   2 ⊃D
    9             A              /    \                  8 ~ ⊃D
   10            ~ B            /      \                 8 ~ ⊃D
   11             ×           ~ C       D                8 ⊃D
                               ×        ×
```

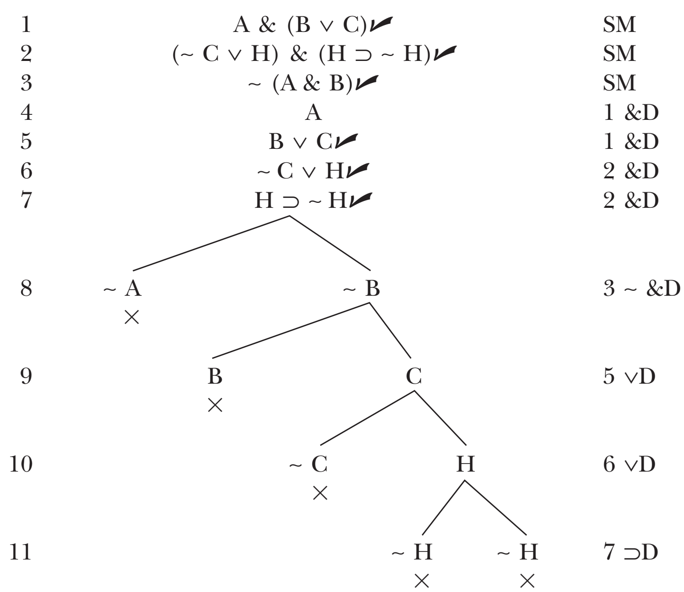
The tree is closed, so the empty set does truth-functionally entail '[(A ⊃ B) ⊃ (C ⊃ D)] ⊃ [C ⊃ (B ⊃ D)]'.

8.a.



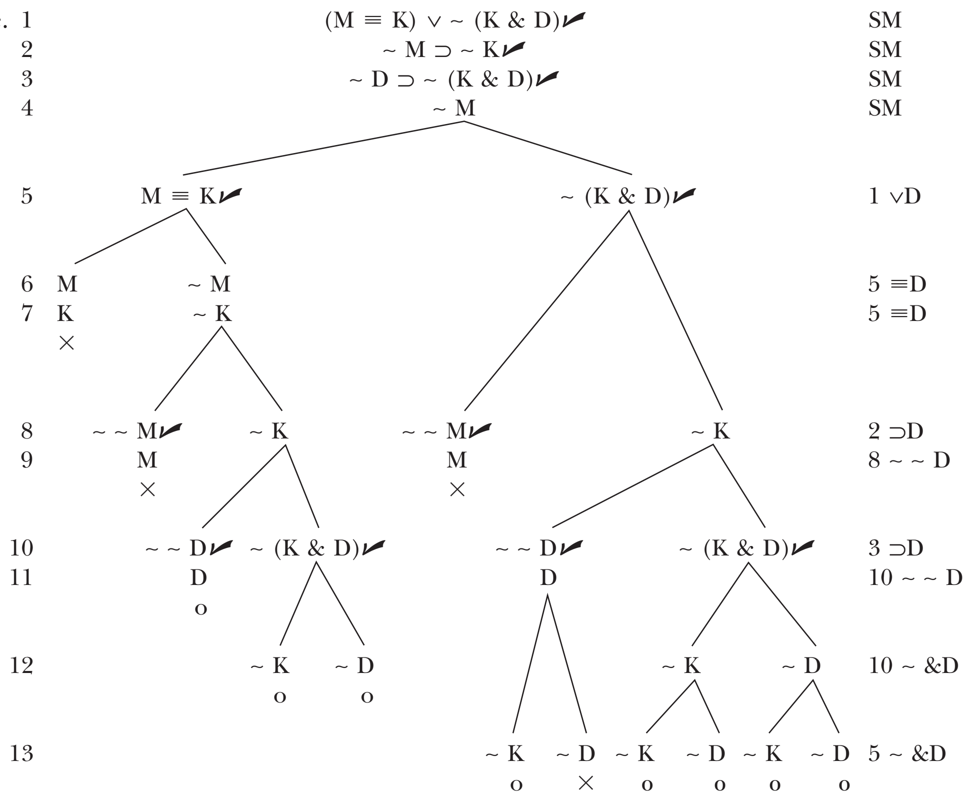
Our truth-tree for the premises and the negation of the conclusion of the argument we are testing is closed. Therefore there is no truth-value assignment on which the premises and the negation of the conclusion are all true, hence no assignment on which the premises are true and the conclusion false. So the argument is truth-functionally valid.

c.



Our truth-tree for the premises and the negation of the conclusion of the argument we are testing is closed. Therefore the argument is truth-functionally valid.

e.



Our truth-tree for the premises and the negation of the conclusion of the argument we are testing is open. Therefore that argument is truth-functionally invalid. The recoverable sets of truth-value assignments are

D	K	M
T	**F**	**F**
F	**F**	**F**

g.	1	B & (H ∨ Z) ✓	SM
	2	~ Z ⊃ K ✓	SM
	3	(B ≡ Z) ⊃ ~ Z ✓	SM
	4	~ K	SM
	5	~ (M & N)	SM
	6	B	1 &D
	7	H ∨ Z	1 &D
	8	~ ~ Z ✓ K ×	2 ⊃D
	9	Z	8 ~ ~ D
	10	~ (B ≡ Z) ✓ ~ Z ×	3 ⊃D
	11	B ~ B	10 ~ ≡D
	12	~ Z × Z ×	10 ~ ≡D

Our truth-tree for the premises and the negation of the conclusion of the argument we are testing is closed. Therefore that argument is truth-functionally valid. Notice that our tree closed before we decomposed the negation of the conclusion. Thus the premises of the argument form a truth-functionally inconsistent set, and therefore those premises and any conclusion constitute a truth-functionally valid argument, even where the conclusion has no atomic components in common with the premises.

i.	1	A & (B ⊃ C) ✓	SM
	2	~ ((A & C) ∨ (A & ~ B)) ✓	SM
	3	A	1 &D
	4	B ⊃ C ✓	1 &D
	5	~ (A & C) ✓	2 ~ ∨D
	6	~ (A & ~ B) ✓	2 ~ ∨D
	7	~ B C	4 ⊃D
	8	~ A × ~ C ~ A × ~ C ×	5 ~ &D
	9	~ A × ~ ~ B ✓	6 ~ &D
	10	B ×	9 ~ ~ D

Our truth-tree for the premise and the negation of the conclusion is closed. Therefore the argument is truth-functionally valid.

```
k.  1                 A ⊃ ~ A✓                               SM
    2               (B ⊃ A) ⊃ B✓                             SM
    3               ~ (A ≡ ~ B)✓                             SM
                     /          \
    4               A            ~ A                         3 ~ ≡D
    5             ~ ~ B✓         ~ B                         3 ~ ≡D
    6               B             /  \                       5 ~ ~ D
                  /    \         /    \
    7   ~ (B ⊃ A)✓      B   ~ (B ⊃ A)✓  B                    2 ⊃D
    8        B         / \       B      ×                    7 ~ ⊃D
    9       ~ A       /   \     ~ A                          7 ~ ⊃D
   10        ×      ~ A   ~ A    ×                           1 ⊃D
                     ×     ×
```

Our truth-tree for the premise and the negation of the conclusion is closed. Therefore the argument is truth-functionally valid.

9.a. In symbolizing the argument we use the following symbolization key:

C: Members of Congress claim to be sympathetic to senior citizens.
M: More money will be collected through social security taxes.
S: The social security system will succeed.
T: Many senior citizens will be in trouble.

Here is our tree for the premises and the negation of the conclusion:

```
1       S ≡ M✓       SM
2        S ∨ T       SM
3      C & ~ M✓      SM
4        ~ ~ S✓      SM
5          S         4 ~ ~ D
6          C         3 &D
7         ~ M        3 &D
          /  \
8       S     ~ S    1 ≡D
9       M     ~ M    1 ≡D
        ×      ×
```

Since our truth-tree is closed, the argument is truth-functionally valid.

c. In symbolizing the argument we use the following symbolization key:

A: The President acts quickly.
C: The President is pressured by senior citizens.
D: Senior citizens will be delighted.
H: The President is pressured by members of the House.
M: The President is pressured by members of the Senate.
S: The social security system will be saved.

Here is our tree for the premises and the negation of the conclusion.

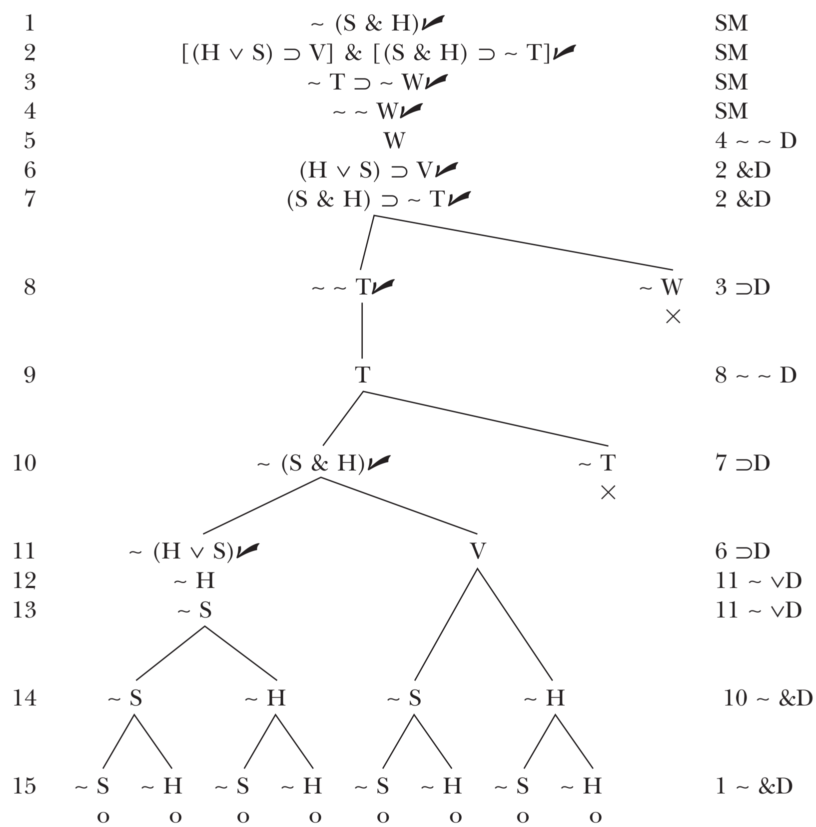
1	$(A \supset S) \& (S \supset D)$ ✓				SM
2	$[(M \vee H) \vee C] \supset A$ ✓				SM
3	$\sim (M \vee H) \& C$ ✓				SM
4	$\sim D$				SM
5	$\sim (M \vee H)$ ✓				3 &D
6	C				3 &D
7	$\sim M$				5 ~ ∨D
8	$\sim H$				5 ~ ∨D
9	$A \supset S$ ✓				1 &D
10	$S \supset D$ ✓				1 &D
11	$\sim [(M \vee H) \vee C]$ ✓	A			2 ⊃D
12	$\sim (M \vee H)$				11 ~ ∨D
13	$\sim C$				11 ~ ∨D
	×				
14		$\sim S$		D	10 ⊃D
				×	
15		$\sim A$	S		9 ⊃D
		×	×		

Since our tree is closed, the argument is truth-functionally valid.

e. In symbolizing the argument we use the following symbolization key:

H: The House of Representatives will pass the bill.
S: The Senate will pass the bill.
T: The President will be pleased.
V: The voters will be pleased.
W: All the members of the White House will be happy.

Here is our tree for the premises and the negation of the conclusion.



Since our truth-tree is open, the argument is truth-functionally invalid. The recoverable sets of truth-value assignments are

H	S	T	W	V
T	**F**	**T**	**T**	**T**
F	**T**	**T**	**T**	**T**
F	**F**	**T**	**T**	**T**
F	**F**	**T**	**F**	**T**

10.a. The first of the following arguments is truth-functionally invalid, the second, truth-functionally valid. In each case the tree for the premise and the conclusion are open. This demonstrates that constructing a tree for the premises of an argument and the conclusion of the argument and finding that the tree has a completed open branch establishes neither that the argument is truth-functionally valid nor that it is truth-functionally invalid.

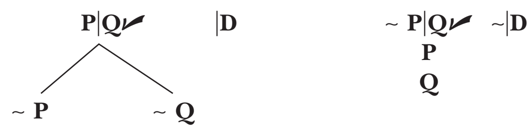
$$\frac{H \vee G}{G} \qquad\qquad \frac{H \,\&\, G}{G}$$

```
1          H ∨ G✓          SM
2            G             SM
            / \
           /   \
3        H       G         1 ∨D
         o       o
```

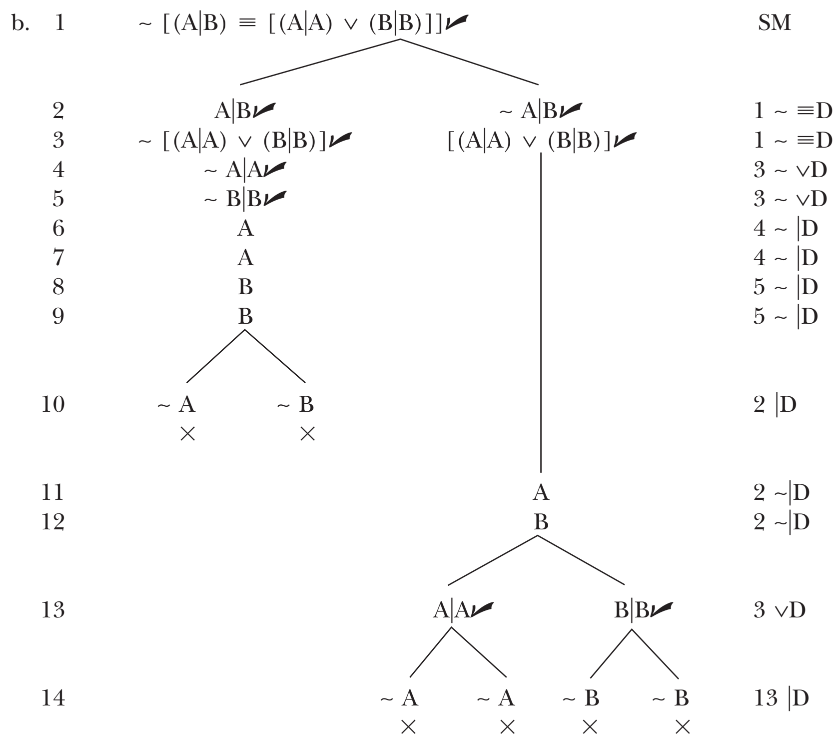
```
1     H & G✓    SM
2       G       SM
3       H       1 &D
4       G       1 &D
        o
```

c. Since constructing a tree for the premises of an argument and the conclusion, whether the tree be open (see answer to a above) or closed (see answer to b above) establishes neither that the argument is truth-functionally valid nor that it is truth-functionally invalid. There is clearly no useful information to be gained by constructing such a tree.

11. a. The need rules are



b.



The truth-tree is closed. Therefore the sentences we are testing are truth-functionally equivalent.