Question 2

1. If I'm coming down with a cold and I exercise, then I'll get worse and feel awful.
2. If I don't exercise, then I'll suffer exercise deprivation and I'll feel awful.

   ∴ If I'm coming down with a cold, then I'll feel awful.

- C = cold
- E = exercise
- W = worse
- A = awful
- D = exercise deprivation

Truth-table Method

C	**E**	**W**	**A**	**D**	**(C · E) ⊃ (W · A)**	**(~E ⊃ (D · A))**	**(C ⊃ A)**
0	0	0	0	0	1	0	1
0	0	0	0	1	1	0	1
0	0	0	1	0	1	0	1
0	0	0	1	1	*1*	*1*	*1*
0	0	1	0	0	1	0	1
0	0	1	0	1	1	0	1
0	0	1	1	0	1	0	1
0	0	1	1	1	*1*	*1*	*1*
0	1	0	0	0	*1*	*1*	*1*
0	1	0	0	1	*1*	*1*	*1*
0	1	0	1	0	*1*	*1*	*1*
0	1	0	1	1	*1*	*1*	*1*
0	1	1	0	0	*1*	*1*	*1*
0	1	1	0	1	*1*	*1*	*1*
0	1	1	1	0	*1*	*1*	*1*
0	1	1	1	1	*1*	*1*	*1*
1	0	0	0	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	0	1	0	1
1	0	0	1	1	*1*	*1*	*1*
1	0	1	0	0	1	0	0

1	0	1	0	1	1	0	0
1	0	1	1	0	1	0	1
1	0	1	1	1	*1*	*1*	*1*
1	1	0	0	0	0	1	0
1	1	0	0	1	0	1	0
1	1	0	1	0	0	1	1
1	1	0	1	1	0	1	1
1	1	1	0	0	0	1	0
1	1	1	0	1	0	1	0
1	1	1	1	0	*1*	*1*	*1*
1	1	1	1	1	*1*	*1*	*1*

Every time the premises are true, the conclusion is also true.

S/I Rules Strictly

1. (C ・ E) ⊃ (W ・ A)
2. (~E ⊃ (D ・ A))

   ∴ (C ⊃ A)

3. assume: ~(C ⊃ A)
    4. ∴ (C ・ ~A) (from 3)
    5. ∴ C (from 4)
    6. ∴ ~A (from 4)
    7. assume: (C ・ E)
        8. ∴ C (from 7)
        9. ∴ E (from 7)
        10. ∴ (W ・ A) (from 1 and 7)
        11. ∴ W (from 10)
        12. ∴ A (from 10)
    8. ∴ ~(C ・ E) (6 contradicts 12)
    9. ∴ (~C ∨ ~E) (from 8)
    10. ∴ ~E (from 5 and 9)
    11. ∴ (D ・ A) (from 2 and 10)
    12. ∴ D (from 11)
    13. ∴ A (from 11)

4. ∴ (C ⊃ A) (6 contradicts with 13)

Resolution proof

1. (C · E) ⊃ (W · A)

- ~(C · E) ∨ (W · A)
- (~C ∨ ~E) ∨ (W · A)
- (~C ∨ ~E ∨ W) · (~C ∨ ~E ∨ A)
- Clauses:
    - (~C ∨ ~E ∨ W)
    - (~C ∨ ~E ∨ A)

2. (~E ⊃ (D · A))

- ~(~E) ∨ (D · A)
- (E ∨ D ∨ A)
- Clause:
    - (E ∨ D ∨ A)

3. ~(C ⊃ A)

- ~(~C ∨ A)
- (C · ~A)
- Clauses:
    - C
    - ~A
- Resolving C with (~C ∨ ~E ∨ W) gives (~E ∨ W)
- Resolving C with (~C ∨ ~E ∨ A) gives (~E ∨ A)
- Resolving ~A with (~E ∨ A) gives ~E
- Resolving ~E with (E ∨ D ∨ A) gives (D ∨ A)
- Resolving ~A with (D ∨ A) gives D
- Resolving D with (E ∨ D ∨ A) gives (E ∨ A)
- Resolving ~E with (E ∨ A) gives A
- Resolving ~A with A gives empty clause

Question 4

1. If President Nixon knew about the massive Watergate cover-up, then he lied to the American people on national television and he should resign.
2. If President Nixon didn't know about the massive Watergate cover-up, then he was incompetently ignorant and he should resign.

   ∴ Nixon should resign.

- K = knew about the massage Watergate cover-up
- L = lied to the American people on national television
- R = should resign
- I = incompletely ignorant

Truth-tabe method

K	L	R	I	(K ⊃ (L · R))	(~K ⊃ (I · R))	R
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	*1*	*1*	*1*
0	1	0	0	1	0	0
0	1	0	1	1	0	0
0	1	1	0	1	0	1
0	1	1	1	*1*	*1*	*1*
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	*1*	*1*	*1*
1	1	1	1	*1*	*1*	*1*

Every time the premises are true, the conclusion is also true.

S/I Rules Strictly

1. (K ⊃ (L · R))

2. (~K ⊃ (I · R)) ∴ R
3. assume: ~R
    4. ∴ ~(L · R) (from 1 and 3)
    5. ∴ ~K (from 1 and 4)
    6. ∴ (I · R) (from 2 and 5)
    7. ∴ I (from 6)
    8. ∴ R (from 6)
4. ∴ R (3 contradicts 8)

Resolution proof

1. (K ⊃ (L · R))

- (~K ∨ (L · R))
- ((~K ∨ L) · (~K ∨ R))
- Clauses:
    - (~K ∨ L)
    - (~K ∨ R)

2. (~K ⊃ (I · R))

- (K ∨ (I · R))
- ((K ∨ I) · (K ∨ R))
- Clauses:
    - (K ∨ I)
    - (K ∨ R)

3. ~R

- Clause:
    - ~R
- Resolving ~R with (~K ∨ R) gives ~K
- Resolving ~K with (K ∨ I) gives I
- Resolving ~K with (K ∨ R) gives R
- Resolving R with ~R gives empty clause

Question 7

Question 7

1. If Michigan either won or tied, then Michigan is going to the Rose Bowl and Gensler is happy.

   ∴ If Gensler isn't happy, then Michigan didn't tie.

- W = Michigan won
- T = Michigan tied
- R = Michigan is going to the Rose Bowl
- H = Gensler is happy

Truth-table method

W	T	R	H	((W ∨ T) ⊃ (R · H))	(~H ⊃ ~T)
0	0	0	0	*1*	*1*
0	0	0	1	*1*	*1*
0	0	1	0	*1*	*1*
0	0	1	1	*1*	*1*
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	*1*	*1*
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	*1*	*1*
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	*1*	*1*

Every time the premises are true, the conclusion is also true.

S/I Rules Strictly

1. ((W ∨ T) ⊃ (R · H))

   ∴ (~H ⊃ ~T)

2. Assume: ~(~H ⊃ ~T)

   3. ∴ (~H · T) (from 2)
   4. ∴ ~H (from 3)
   5. ∴ T (from 3)
   6. ∴ (W ∨ T) (from 1 and 5)
   7. ∴ (R · H) (from 1 and 6)
   8. ∴ R (from 7)
   9. ∴ H (from 7)

3. ∴ (~H ⊃ ~T) (4 contradicts 9)

Resolution proof

1. ((W ∨ T) ⊃ (R · H))

- (~(W ∨ T) ∨ (R · H))
- ((~W · ~T) ∨ (R · H))
- ((~W ∨ R) · (~W ∨ H) · (~T ∨ R) · (~T ∨ H))
- Clauses:
  - (~W ∨ R)
  - (~W ∨ H)
  - (~T ∨ R)
  - (~T ∨ H)

2. ~(~H ⊃ ~T)

- ~(~H ∨ ~T)
- (H · T)
- Clauses:
  - H
  - T
- Resolving T with (~T ∨ R) gives R
- Resolving H with (~T ∨ H) gives ~T
- Resolving ~T with T gives empty clause

Connectives

Given the standard set of connectives is adequate, prove that {~, V} is also adequate

- (P · Q) = ~(~P ∨ ~Q)
- (P ⊃ Q) = (~P ∨ Q)
- (P ≡ Q) = ((P ⊃ Q) · (Q ⊃ P)) = ((~P ∨ Q) · (~Q ∨ P)) = ~((~P ∨ Q) ∨ (~Q ∨ P))

Prove that {V} alone is not adequate for propositional logic

- Assume that {∨} is adequate. Then we must be able to express all the expresions above only using {∨}
- { ~ } flips the truth value of a statement. Since {∨} is a monotonic operator, you can't flip a truth value. For example, it's impossible to show that P = 0 without { ~ }. Therefore {∨} can't be used in place of {~}.
- For (P · Q) = ~(~P ∨ ~Q), we can't write this with just {∨} for the same reason as above.
- I could write the same thing for all the expressions above. Therefore, {∨} alone is not adequate for propositional logic.