**Name:** Mohamed Elkholy

**HW7**

**1. SAT Modeling:**

**1.1 Scenario A:**

1. A: Ice cream is selected

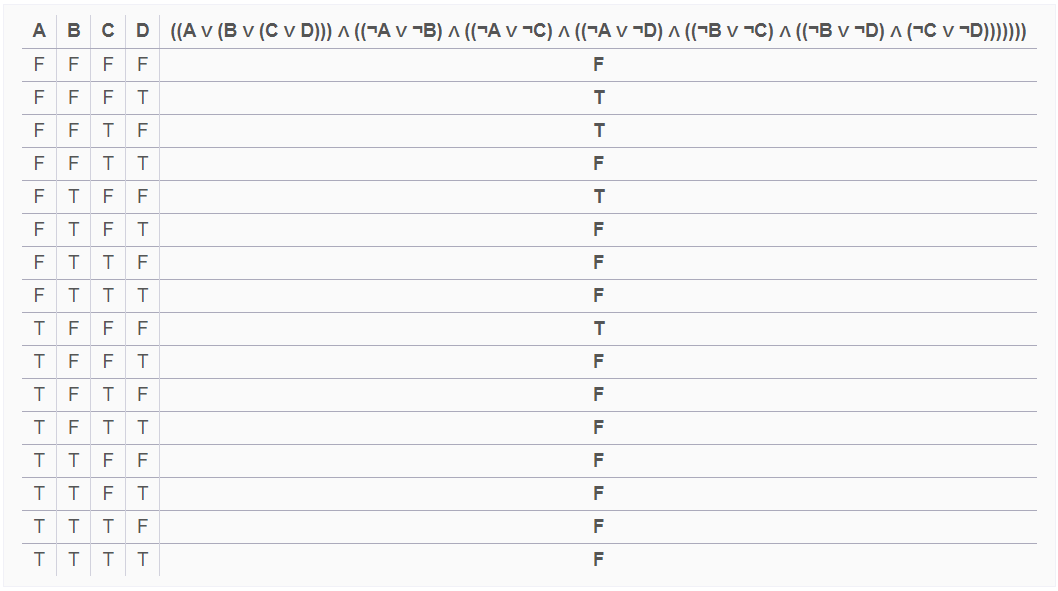
B: Fruit bowl is selected

C: Cake is selected

D: Pie is selected

1. Exactly one dessert must be selected

(A ∨ B ∨ C ∨ D) ∧ (¬A ∨ ¬B) ∧ (¬A ∨ ¬C) ∧ (¬A ∨ ¬D) ∧ (¬B ∨ ¬C) ∧ (¬B ∨ ¬D) ∧ (¬C ∨ ¬D)



1. (A ∨ B ∨ C ∨ D): At least one dessert is selected

(¬A ∨ ¬B): Ice cream and fruit bowl cannot be selected together

(¬A ∨ ¬C): Ice cream and cake cannot be selected together

(¬A ∨ ¬D): Ice cream and pie cannot be selected together

(¬B ∨ ¬C): Fruit bowl and cake cannot be selected together

(¬B ∨ ¬D): Fruit bowl and pie cannot be selected together

(¬C ∨ ¬D): Cake and pie cannot be selected together

1. Yes, the sentence is satisfiable. The CNF formula ensures that at least one dessert is selected.

**1.2 Scenario B:**

1. D**i**: Damon is available on a particular day i : i ∈ {M, Tu, W, Th, F}.

E**i**: Enriqueis available on a particular day i : i ∈ {M, Tu, W, Th, F}.

L**i**: Loisis available on a particular day i : i ∈ {M, Tu, W, Th, F}.

PA**i**​: The paper is done on a particular day i : i ∈ {M, Tu, W, Th, F}.

PR**i**​: The paper is done on a particular day i : i ∈ {M, Tu, W, Th, F}.

1. ¬D**M** ∧ (L**M** ∧ L**Tu** ∧ L**W**) ∧ (E**i**​ v E**i+1**​) ∧ (¬Ed v ¬Ed+1) ∧ (PA**i**​ ∧ ¬PR**j**) : i < j
2. ¬D**M**: Damon cannot meet on Monday

L**M** ∧ L**Tu** ∧ L**W**: Lois wants to complete the presentation on or before Wednesday

(PA**i**​ ∧ ¬PR**j**) : i < j : Damon wants to complete the paper before the presentation and not both on the same day

(E**i**​ XOR E**i+1**) = (E**i**​ v E**i+1**​) ∧ (¬Ed v ¬Ed+1): Enrique can meet any day but cannot meet on two consecutive days

1. No.

First, we cannot choose Monday. Also, the presentation cannot be done on either Th or F.

Since the paper must be completed before the presentation, the paper cannot be done on W, Th, or F.

As a result, they can only work on the paper on Tu, while working on the Presentation on either Tu or W which does not satisfy the conditions that do not allow working on them on the same day or two consecutive days.

**1.3 Scenario C:**

1. NE\_R: Nebraska is red

NE\_G: Nebraska is green

NE\_B: Nebraska is blue

IA\_R: Iowa is red

IA\_G: Iowa is green

IA\_B: Iowa is blue

KS\_R: Kansas is red

KS\_G: Kansas is green

KS\_B: Kansas is blue

MO\_R: Missouri is red

MO\_G: Missouri is green

MO\_B: Missouri is blue

1. (NE\_R v NE\_B v NE\_G) ∧ (¬NE\_R v ¬NE\_B) ∧ (¬NE\_R v ¬NE\_G) ∧ (¬NE\_B v ¬NE\_G) ∧

(IA\_R v IA\_B v IA\_G) ∧ (¬IA\_R v ¬IA\_B) ∧ (¬IA\_R v ¬IA\_G) ∧ (¬IA\_B v ¬IA\_G) ∧

(KS\_R v KS\_B v KS\_G) ∧ (¬KS\_R v ¬KS\_B) ∧ (¬KS\_R v ¬KS\_G) ∧ (¬KS\_B v ¬KS\_G) ∧

(MO\_R v MO\_B v MO\_G) ∧ (¬MO\_R v ¬MO\_B) ∧ (¬MO\_R v ¬MO\_G) ∧ (¬MO\_B v ¬MO\_G) ∧

(¬NE\_R v ¬IA\_R) ∧ (¬NE\_G v ¬IA\_G) ∧ (¬NE\_B v ¬IA\_B) ∧

(¬NE\_R v ¬MO\_R) ∧ (¬NE\_G v ¬MO\_G) ∧ (¬NE\_B v ¬MO\_B) ∧

(¬NE\_R v ¬KS\_R) ∧ (¬NE\_G v ¬KS\_G) ∧ (¬NE\_B v ¬KS\_B) ∧

(¬IA\_R v ¬MO\_R) ∧ (¬IA\_G v ¬MO\_G) ∧ (¬IA\_B v ¬MO\_B) ∧

(¬MO\_R v ¬KS\_R) ∧ (¬MO\_G v ¬KS\_G) ∧ (¬MO\_B v ¬KS\_B)

1. (NE\_R v NE\_B v NE\_G): Ensures that NE is colored

(IA\_R v IA\_B v IA\_G): Ensures that IA is colored

(KS\_R v KS\_B v KS\_G): Ensures that KS is colored

(MO\_R v MO\_B v MO\_G): Ensures that MO is colored

(¬NE\_R v ¬NE\_B) ∧ (¬NE\_R v ¬NE\_G) ∧ (¬NE\_B v ¬NE\_G): NE must be colored with exactly one color

(¬IA\_R v ¬IA\_B) ∧ (¬IA\_R v ¬IA\_G) ∧ (¬IA\_B v ¬IA\_G): IA must be colored with exactly one color

(¬KS\_R v ¬KS\_B) ∧ (¬KS\_R v ¬KS\_G) ∧ (¬KS\_B v ¬KS\_G): KS must be colored with exactly one color

(¬MO\_R v ¬MO\_B) ∧ (¬MO\_R v ¬MO\_G) ∧ (¬MO\_B v ¬MO\_G): MO must be colored with exactly one color

(¬NE\_R v ¬IA\_R) ∧ (¬NE\_G v ¬IA\_G) ∧ (¬NE\_B v ¬IA\_B): NE and IA cannot have the same color

(¬NE\_R v ¬MO\_R) ∧ (¬NE\_G v ¬MO\_G) ∧ (¬NE\_B v ¬MO\_B): NE and MO cannot have the same color

(¬NE\_R v ¬KS\_R) ∧ (¬NE\_G v ¬KS\_G) ∧ (¬NE\_B v ¬KS\_B): NE and KS cannot have the same color

(¬IA\_R v ¬MO\_R) ∧ (¬IA\_G v ¬MO\_G) ∧ (¬IA\_B v ¬MO\_B): IA and MO cannot have the same color

(¬MO\_R v ¬KS\_R) ∧ (¬MO\_G v ¬KS\_G) ∧ (¬MO\_B v ¬KS\_B): MO and KS cannot have the same color

1. No. Since NE shares borders with the other four states, the four states must have different colors. However, there are only three colors, so the four states cannot have different colors.

**2. Chapter 7, Exercise 1:**

KB:

(W1,3 ∧ ¬W2,2) v (¬W1,3 ∧ W2,2)

(¬W2,2 ∧ ¬W3,1)

(W1,3 ∧ ¬W2,2 ∧ ¬W3,1)

(P2,2 ∧ P3,1)

(¬P1,3 ∧ ¬P2,2)

(P3,1 ∧ ¬P3,1 ∧ ¬P2,2)

| **W1,3 , W2,2 or W3,1** | **P1,3** | **P2,2** | **P3,1** | **(W1,3 ∧ ¬W2,2 ∧ ¬W3,1)** | **(P3,1 ∧ ¬P3,1 ∧ ¬P2,2)** | **KB** | **a2** | **a3** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| W1,3 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| **W1,3** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** |
| W1,3 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| W1,3 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| W1,3 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| W1,3 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| W1,3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| W1,3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| W2,2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| W2,2 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| W2,2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| W2,2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| W2,2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| W2,2 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| W2,2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| W2,2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| W3,1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| W3,1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| W3,1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| W3,1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| W3,1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| W3,1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| W3,1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| W3,1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

We can notice that when KB is True, a2 is true (KB |= a2). Therefore, a2 = "There is not pit in [2,2]."

We can also notice that when KB is True, a3 is true (KB |= a3). Therefore, a3 = "There is a wumpus in [1,3]."

**3. Chapter 7, Exercise 2:**

P1: The unicorn is mythical

P2: The unicorn is mortal

P3: The unicorn is a mammal

P4: The unicorn is horned

P5: The unicorn is magical

P1 →¬P2 **(1)**

¬P1 → (P2 P3) **(2)**

(¬P2 ∨ P3) → P4 **(3)**

P4 → P5 **(4)**

From (1): P1 →¬P2 ≡ ¬P1 v ¬P2 **(5)**

From (2):

¬P1 → (P2 ∧ P3) ≡ (P1 ∨ P2) ∧ (P1 ∨ P3)

Therefore, (P1 ∨ P2) **(6)**

(P1 ∨ P3) **(7)**

By Resolution on (5) and (7): (¬P2 ∨ P3) **(8)**

By Modus Ponens on (3) and (8): P4 is True.(The unicorn is horned) **(9)**

By Modus Ponens on (4) and (9): P5 is True. (The unicorn is magical) **(10)**

According to the knowledge base, we cannot explicitly identify whether P1 is True or False.

**4. Chapter 7, Exercise 9:**

1. The sentence is True 3 times for each combination of two variables. Therefore, there are 3\*4=12 models.
2. The sentence is False only if A, B, C, and D are all True which happens once. Therefore, there are 2^4 - 1 = 15 models.
3. The sentence is True when A is True and B is False which makes (A ⇒ B) False. Therefore, there is no model.

**5. Truth Tables:**

| **p** | **q** | **p ∧ q** | **¬p** | **¬q** | **¬p ∨ ¬q** | **¬(¬p ∨ ¬q)** | **(p ∧ q) → ¬(¬p ∨ ¬q)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| T | T | T | F | F | F | T | **T** |
| T | F | F | F | T | T | F | **T** |
| F | T | F | T | F | T | F | **T** |
| F | F | F | T | T | T | F | **T** |

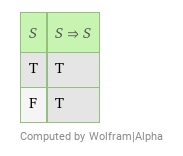
1. M = Mary, S = Susy

| **M** | **S** | **M → S** | **[M ∧ (M → S)]** | **[M ∧ (M → S)] → S** |
| --- | --- | --- | --- | --- |
| T | T | T | T | **T** |
| T | F | F | F | **T** |
| F | T | T | F | **T** |
| F | F | T | F | **T** |

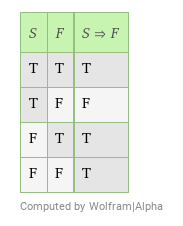
| **a** | **b** | **a ∧ b** | **b → (a ∧ b)** | **a → [b → (a ∧ b)]** |
| --- | --- | --- | --- | --- |
| T | T | T | T | **T** |
| T | F | F | T | **T** |
| F | T | F | F | **T** |
| F | F | F | T | **T** |

| **a** | **b** | **c** | **a → b** | **b → c** | **a → c** | **[(b → c) → (a → c)]** | **(a → b) → [(b → c) → (a → c)]** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| T | T | T | T | T | T | T | **T** |
| T | T | F | T | F | F | T | **T** |
| T | F | T | F | T | T | T | **T** |
| T | F | F | F | T | F | F | **T** |
| F | T | T | T | T | T | T | **T** |
| F | T | F | T | F | T | T | **T** |
| F | F | T | T | T | T | T | **T** |
| F | F | F | T | T | T | T | **T** |

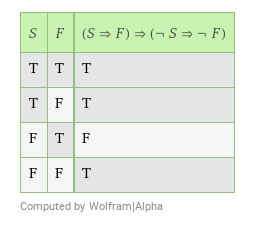
**6. Chapter 7, Exercise 12:**



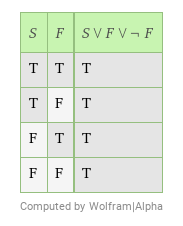
It is a tautology, so it is valid



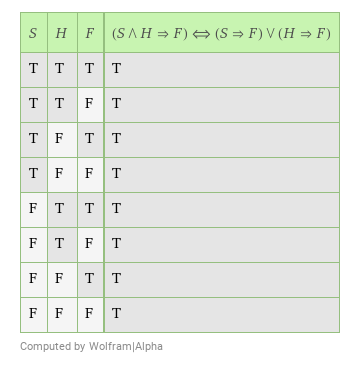
It is not valid but satisfiable.



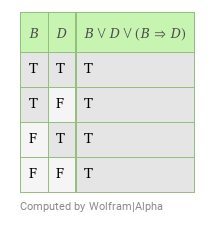
It is not valid but satisfiable.



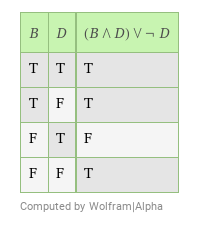
It is a tautology, so it is valid



It is a tautology, so it is valid

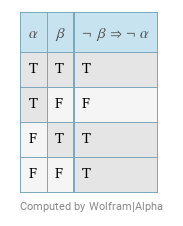
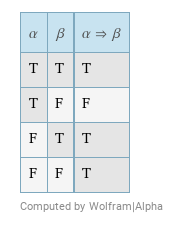


It is a tautology, so it is valid

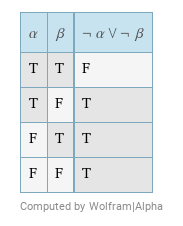
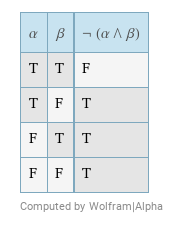


It is not valid but satisfiable.

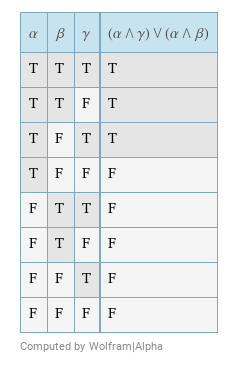
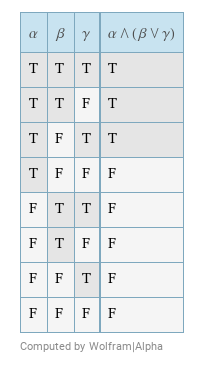
**7. Logical Equivalences:**



The two truth tables are equivalent.



The two truth tables are equivalent.



The two truth tables are equivalent.

**8. Chapter 7, Exercise 28:**

1. (¬X1,2 ∨ ¬X2,1 ∨ ¬X2,2) ∧ (X1,2 ∨ X2,1) ∧ (X1,2 ∨ X2,2) ∧ (X2,1 ∨ X2,2)
2. ⋀ (m,n) neighbors of (i,j) (if k of n neighbors contain mines, then V (m,n) neighbors of (i,j) X(m,n))
   1. Add the relevant clauses to the CNF based on the known information about adjacent mines.
   2. Use DPLL to determine whether the CNF is satisfiable.
   3. If satisfiable, it means the square can contain a mine, and the agent can probe cautiously; if unsatisfiable, the square is safe to probe.
3. The total number of possible clauses is M\*N. The CNF size grows linearly with both M and N.
4. The conclusions derived by the method in part (3) are generally not altered when the global constraint is taken into account. The global constraint might limit the number of mines in the entire grid, but the local constraints around a specific square remain valid.
5. Consider an N × 1 board where N is large.

Example 1: Chain of Mines

Consider a board with mines placed in a consecutive sequence, such as [M, M, M,..., M, M]. In this case, if you probe a square in the middle, like the square at position (1, 1), and it turns out to be clear, you can deduce that all neighboring squares are also clear. Similarly, if it contains a mine, you can deduce that all neighboring squares are mines.

Example 2: Alternating Mines

Consider a board with mines placed in an alternating pattern such as [M, \_, M, \_, M, \_,…, M, \_, M]. If you probe the first square and find it clear (\_), you can deduce that the second and all subsequent even-indexed squares are clear. Similarly, if you find a mine in the first square, you can deduce that the second and all subsequent odd-indexed squares contain mines.

Example 3: Separated Mines

Consider a board with mines separated by clear squares such as [\_, M, \_, M, \_…, \_, M, \_]. If you probe the first square and find it clear (\_), you can deduce that all the squares with mines are far-distant from the first square. Conversely, if you find a mine in the first square, you can deduce the location of mines among the far-distant squares.

**9. Proofs:**

1. q ∧ (r ∧ p): Given

2. t → v: Given

3. v → ¬p: Given

4. t → ¬p: By combining statements 2 and 3 using the property of implications, if t implies v, and v implies ¬p. Then, t implies ¬¬p

5. (r ∧ p): From the given assumption q ∧ (r ∧ p)

6. r: From the given assumption q ∧ (r ∧ p)

7. p: From the given assumption q ∧ (r ∧ p)

8. ¬¬p: Double negation of (7)

9. ¬t: From the contrapositive of (4), then ¬¬p → ¬t

10. ¬t ∧ r: By combining (6) and (9)

1. 1. (¬p v q) ∧ (¬p v r): From the given assumption p → (q ∧ r) ≡ ¬p v (q ∧ r)

2. (¬q v s): From the given assumption q → s

3. (¬r v t): From the given assumption r → t

4. (¬p v s): By resolution on (2) and (¬p v q) from (1)

5. (¬p v t): By resolution on (3) and (¬p v r) from (1)

6. (¬p v s) ∧ (¬p v t): By combining (4) and (5)

7. p → (s ∧ t): From (6) (¬p v s) ∧ (¬p v t) ≡ ¬p v (s ∧ t)

1. 1. ¬(¬p ∧ q): Given

2. p → (¬t ∨ r): Given

3. q: Given

4. t: Given

5. ¬r: Negation of Conclusion

6. (p v ¬q): By applying De Morgan’s Law on (1)

7. q → p: From (6)

8. p: From (6) since q is True from (3)

9. (¬t ∨ r): From (2) since p is True from (8)

10. r: From (9) since t is True from (4)

11. r: (10) contradicts with (5)