problemSet4

Com S 472/572 Spring 2022 Problem Set 4 (65 pts)

Due at 11:59pm Thursday, Mar 24

All exercises are from https://aimacode.github.io/aima-exercises/

6.1 (5 pts)

How many solutions are there for the map-coloring problem in Figure 6.1? How many solutions if four colors are allowed? Two colors?

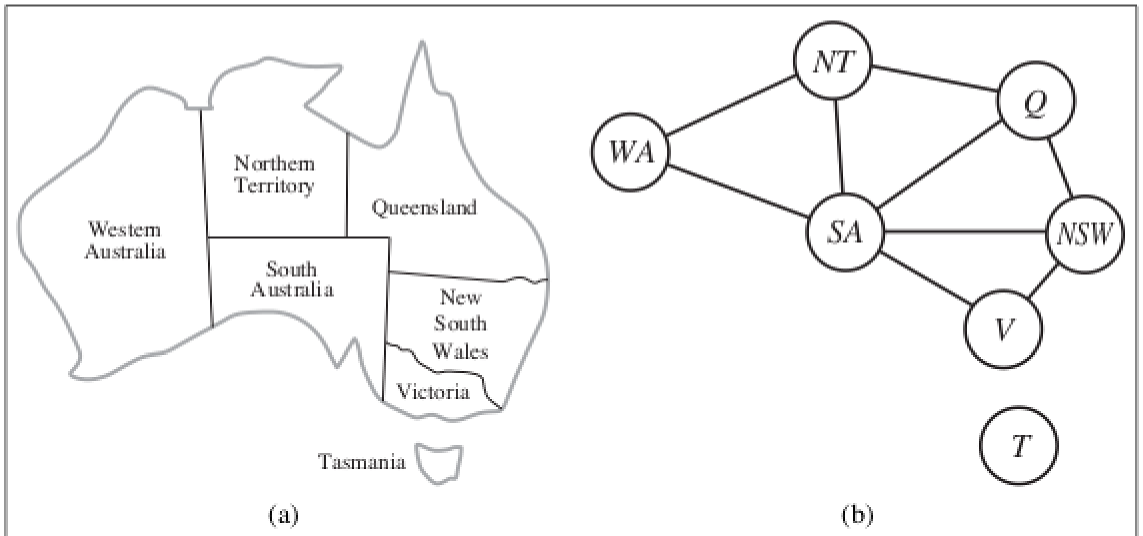

Figure 6.1 (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.

3 Colors:

- T gets all 3 choices
- SA gets 3 colors to choose from
- WA gets 2 choices bc can't choose SA's color
- NT gets 1 choices bc can't choose WA or SA

- Q gets 1 choices bc can't choose NT or SA
- NSW gets 1 choices bc can't choose Q or SA
- V gets 1 choices bc can't choose NSW or SA

3 x 3 x 2 x 1 x 1 x 1 x 1 = 18

4 Colors:

- T gets all 4 choices
- SA gets 4 colors to choose from
- WA gets 3 choices bc can't choose SA's color
- NT gets 2 choices bc can't choose WA or SA
- Q gets 2 choices bc can't choose NT or SA
- NSW gets 2 choices bc can't choose Q or SA
- V gets 2 choices bc can't choose NSW or SA

4 x 4 x 3 x 2 x 2 x 2 x 2 = 768

2 Colors:

No solutions, because if WA is color 1, SA has to be the color 2, then NT can be neither color 1 or color 2.

6.6 (10 pts)

Show how a single ternary constraint such as $A + B = C$ can be turned into three binary constraints by using an auxiliary variable. You may assume finite domains.

(*Hint:* Consider a new variable that takes on values that are pairs of other values, and consider constraints such as X is the first element of the pair Y.)

Next, show how constraints with more than three variables can be treated similarly.

Finally, show how unary constraints can be eliminated by altering the domains of variables. This completes the demonstration that any CSP can be transformed into a CSP with only binary constraints.

- $consider\ D(firstElement, secondElement)$
- i. $A = D(firstElement)$
- ii. $B = D(secondElement)$
- iii. $D(value) = D(sumOfElements)$
- $Therefore\ D = C$

6.11 (10 pts)

- partial assignment {WA = green, V = red}

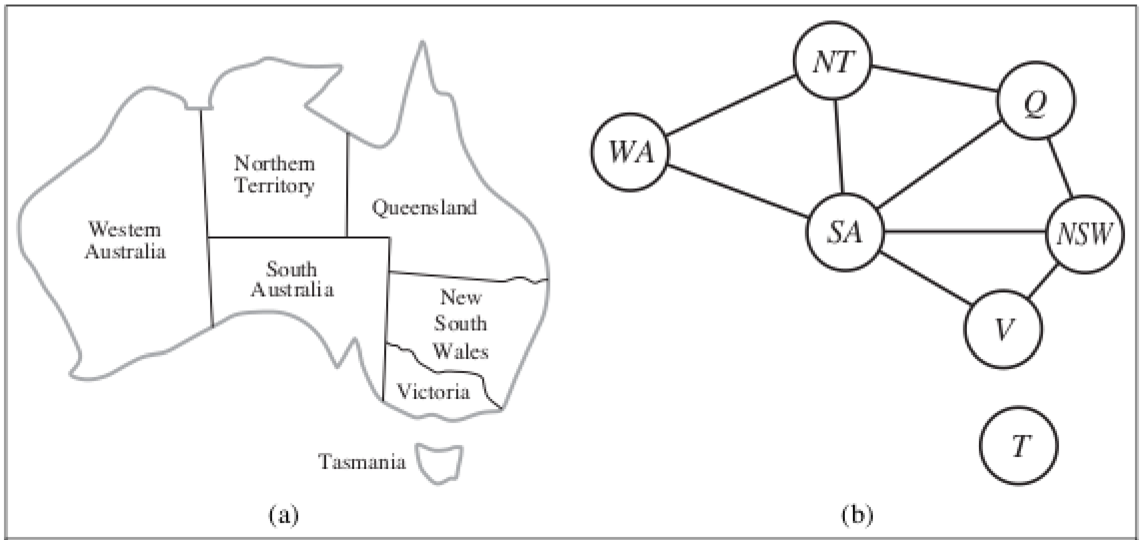

Figure 6.1 (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.

- Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of the partial assignment green, red for the problem shown in Figure 6.1.

available colors: Green, Red, Blue

```
Initial State:
queue =
    {SA,WA}
    {NSW,SA}
    {SA,NSW}
    {V,SA}
    {NT,WA}
    {NT,SA}
```

```
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}

        WA={green}
        SA={red,green,blue}
        T={red,green,blue}
        NT={red,green,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 1:
    SA's color domain changed
    popped =
        {SA,WA}
    queue =
        {NSW,SA}
        {SA,NSW}
        {V,SA}
        {NT,WA}
        {NT,SA}
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,green,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 2:
    no color domain changed
    popped =
        {NSW,SA}
    queue =
        {SA,NSW}
        {V,SA}
        {NT,WA}
        {NT,SA}
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,green,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 3:
    no color domain changed
    popped =
        {SA,NSW}
    queue =
        {V,SA}
        {NT,WA}
        {NT,SA}
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,green,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 4:
    no color domain changed
    popped =
        {V,SA}
    queue =
        {NT,WA}
        {NT,SA}
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,green,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 5:
    NT's color domain changed
    popped =
        {NT,WA}
    queue =
        {NT,SA}
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 6:
    no color domain changed
    popped =
        {NT,SA}
    queue =
        {SA,Q}
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 7:
    no color domain changed
    popped =
        {SA,Q}
    queue =
        {WA,SA}
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 8:
    no color domain changed
    popped =
        {WA,SA}
    queue =
        {Q,SA}
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 9:
    no color domain changed
    popped =
        {Q,SA}
    queue =
        {Q,NSW}
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 10:
    no color domain changed
    popped =
        {Q,NSW}
    queue =
        {NT,Q}
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 11:
    no color domain changed
    popped =
        {NT,Q}
    queue =
        {Q,NT}
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 12:
    no color domain changed
    popped =
        {Q,NT}
    queue =
        {NSW,Q}
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 13:
    no color domain changed
    popped =
        {NSW,Q}
    queue =
        {NSW,V}
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={red,green,blue}
        V={red}
    Step 14:
    NSW's color domain changed
    popped =
        {NSW,V}
    queue =
        {SA,NT}
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={green,blue}
        V={red}
    Step 15:
    no color domain changed
    popped =
        {SA,NT}
    queue =
        {V,NSW}
        {SA,V}
        {WA,NT}
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}

        WA={green}
        SA={red,blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={green,blue}
    V={red}
Step 16:
no color domain changed
popped =
    {V,NSW}
queue =
    {SA,V}
    {WA,NT}
    {NT,SA}
    {Q,SA}
    {NSW,SA}
    {V,SA}
    {SA,NT}
    {Q,NT}
    {SA,NSW}
    {Q,NSW}

    WA={green}
    SA={red,blue}
    T={red,green,blue}
    NT={red,blue}
    Q={red,green,blue}
    NSW={green,blue}
    V={red}
Step 17:
SA's color domain changed
popped =
    {SA,V}
queue =
    {WA,NT}
    {NT,SA}
    {Q,SA}
    {NSW,SA}
    {V,SA}
    {SA,NT}
    {Q,NT}
    {SA,NSW}
    {Q,NSW}
    {NSW,SA}
    {Q,SA}
    {NT,SA}
    {WA,SA}

    WA={green}
    SA={blue}
```

```
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={green,blue}
        V={red}
    Step 18:
    no color domain changed
    popped =
        {WA,NT}
    queue =
        {NT,SA}
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red,blue}
        Q={red,green,blue}
        NSW={green,blue}
        V={red}
    Step 19:
    NT's color domain changed
    popped =
        {NT,SA}
    queue =
        {Q,SA}
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={red,green,blue}
        NSW={green,blue}
        V={red}
    Step 20:
    Q's color domain changed
    popped =
        {Q,SA}
    queue =
        {NSW,SA}
        {V,SA}
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}
        {NSW,Q}
        {NT,Q}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={red,green}
        NSW={green,blue}
        V={red}
    Step 21:
    NSW's color domain changed
    popped =
        {NSW,SA}
    queue =
        {V,SA}
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}
        {NSW,Q}
        {NT,Q}
        {V,NSW}
        {Q,NSW}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={red,green}
        NSW={green}
        V={red}
    Step 22:
    no color domain changed
    popped =
        {V,SA}
    queue =
        {SA,NT}
        {Q,NT}
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}
        {NSW,Q}
        {NT,Q}
        {V,NSW}
        {Q,NSW}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={red,green}
        NSW={green}
        V={red}
    Step 23:
    no color domain changed
    popped =
        {SA,NT}
    queue =
        {Q,NT}
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}
        {NSW,Q}
        {NT,Q}
        {V,NSW}
        {Q,NSW}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={red,green}
        NSW={green}
        V={red}
    Step 24:
    Q's color domain changed
    popped =
        {Q,NT}
    queue =
        {SA,NSW}
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}
        {NSW,Q}
        {NT,Q}
        {V,NSW}
        {Q,NSW}
        {NSW,Q}
        {SA,Q}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={green}
        NSW={green}
        V={red}
    Step 25:
    no color domain changed
    popped =
        {SA,NSW}
    queue =
        {Q,NSW}
        {NSW,SA}
        {Q,SA}
        {NT,SA}
        {WA,SA}
        {Q,NT}
        {WA,NT}
        {NSW,Q}
        {NT,Q}
        {V,NSW}
        {Q,NSW}
        {NSW,Q}
        {SA,Q}

        WA={green}
        SA={blue}
        T={red,green,blue}
        NT={red}
        Q={green}
        NSW={green}
        V={red}
    Step 26:
    Terminated: no color possible
    popped =
        {Q,NSW}
    queue =
        {NSW,SA}
        {Q,SA}
        {NT,SA}
```

```
{WA,SA}
{Q,NT}
{WA,NT}
{NSW,Q}
{NT,Q}
{V,NSW}
{Q,NSW}
{NSW,Q}
{SA,Q}

WA={green}
SA={blue}
T={red,green,blue}
NT={red}
Q={green}
NSW={green}
V={red}
```

6.20 (2+2+2+2=8 pts)

Consider the problem of tiling a surface (completely and exactly covering it) with n dominoes (2×1 rectangles). The surface is an arbitrary edge-connected (i.e., adjacent along an edge, not just a corner) collection of 2n 1×1 squares (e.g., a checkerboard, a checkerboard with some squares missing, a 10×1 row of squares, etc.).

1. Formulate this problem precisely as a CSP where the dominoes are the variables.

- dominoes are all represented as D_1 D_2 D_3 … D_n
- all n dominoes has 2 X values (X_1, X_2), and 2 Y values (Y_1, Y_2), each which are the coordinates of the first piece of the domino and the second piece of the domino
- all n dominoes must either have the same X_1 and X_2 value, with Y_1 and Y_2 being one apart, or vice versa
- all n dominoes must never have any overlapping X Y pairs with any other X Y pairs of any other domino
- all n dominoes' X Y pairs must be contained within the set of all coordinates in the surface
- all coordinates in the surface must be contained within the set of all n dominoes' X Y pairs

2. Formulate this problem precisely as a CSP where the squares are the variables, keeping the state space as small as possible. (*Hint:* does it matter which particular domino goes on a given pair of squares?)

- squares are all represented as S_1 S_2 S_3 … S_n
- all squares must have 1 partner square that is directly adjacent to it
- all partner squares much point to each other
  - ex: if A is a partner to B, B's partner must be A

3. Construct a surface consisting of 6 squares such that your CSP formulation from part (b) has a *tree-structured* constraint graph.

Domino Graph:

a
b
c
d
e
f

Constraint Graph:

```
a
|
b
|
c
|
d
|
e
|
f
```

4. Describe exactly the set of solvable instances that have a tree-structured constraint graph.

- pick edge and remove both nodes and every edge connecting to each of the nodes
- if it is possible to repeat step 1 until there are no nodes left in the graph, then it is solvable

7.4 (12 X 1 = 12 pts)

Which of the following are correct?

1. False $\models$ True.

> Correct

2. True $\models$ False.

> Incorrect

3. $(A \land B) \models (A \Leftrightarrow B)$.

> Correct

4. $A \Leftrightarrow B \models A \lor B$.

- $(A \Rightarrow B) \land (B \Rightarrow A) \models A \lor B$.
- $(\neg A \lor B) \land (\neg B \lor A) \models A \lor B$.
- Counter Example:
- A = False
- B = False

> Incorrect

5. $A \Leftrightarrow B \models \neg A \lor B$.

- $(A \Rightarrow B) \land (B \Rightarrow A) \models \neg A \lor B$.
- $(\neg A \lor B) \land (\neg B \lor A) \models \neg A \lor B$.

- Every time the first statement is true the first part of the and must also be true, meaning that this is true

> Correct

6. $(A \wedge B) \implies C \models (A \Rightarrow C) \vee (B \implies C)$.

- $\neg(A \wedge B) \vee C \models (\neg A \vee C) \vee (\neg B \vee C)$.
- $\neg A \vee \neg B \vee C \models (\neg A \vee C) \vee (\neg B \vee C)$.
- $\neg A \vee \neg B \vee C \models \neg A \vee C \vee \neg B \vee C$.
- $\neg A \vee \neg B \vee C \models \neg A \vee \neg B \vee C$.

> Correct

7. $(C \vee (\neg A \wedge \neg B)) \equiv ((A \implies C) \wedge (B \implies C))$.

- $(C \vee (\neg A \wedge \neg B)) \equiv (\neg A \vee C) \wedge (\neg B \vee C)$.
- $(C \vee (\neg A \wedge \neg B)) \equiv (\neg A \wedge \neg B) \vee C$.

> Correct

8. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$.

- Every time the first statement is true the first part of the and must also be true, meaning that this is true

> Correct

9. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B) \wedge (\neg D \vee E)$.

- Counter Example:
- A: True
- B: True
- C: False
- D: True
- E: False

> Incorrect

10. $(A \lor B) \land \neg(A \implies B)$ is satisfiable.

- $(A \lor B) \land \neg(\neg A \lor B)$
- $(A \lor B) \land (A \land \neg B)$
- $(A \lor B) \land (A \land \neg B)$
- A = True
- B = False

> Correct

11. $(A \Leftrightarrow B) \land (\neg A \lor B)$ is satisfiable.

- $(A \implies B) \land (B \implies A) \land (\neg A \lor B)$.
- $(\neg A \lor B) \land (\neg B \lor A) \land (\neg A \lor B)$.
- A = False
- B = False

> Correct

12. $(A \Leftrightarrow B) \Leftrightarrow C$ has the same number of models as $(A \Leftrightarrow B)$ for any fixed set of proposition symbols that includes A, B, C.

> Incorrect

7.6a, b, e (4+3+3 = 10 pts)

Prove each of the following assertions: a. a is valid if and only if True$\models$a.

True$\models$False is false

True$\models$True is True

therefore, a must always be true for the statement to be true

Therefore a must be valid

b. For any a, False$\models$a.

- False⊨False is true.
- False⊨True is true.
- Therefore, False⊨a is always true because a can be either False or True and the statement will still be true

e. a⊨B if and only if the sentence (a∧¬B) is unsatisfiable.

- assume a = True, because if a = False, (a∧¬B) is unsatisfiable.
- True⊨B must also be true
- B must also be true by definition of ⊨
- Therefore, (a∧¬B) cannot be satisfied
- Therefore, (a∧¬B) is unsatisfiable.

7.7 (2+4+4 = 10 pts)

Prove, or find a counterexample to, each of the following assertions:

1. If a⊨y or B⊨y (or both) then (a∧B)⊨y

- assume 2 steps a⊨y or B⊨y, and prove (a∧B)⊨y

2. If (a∧B)⊨y then a⊨y or B⊨y (or both).

- Counter Example: y=(a∧B)

3. If a⊨(B∨y) then a⊨B or a⊨y (or both).

- Counter Example: a=(B∨y)