hw

# 1

Prove equivalence between $P \Rightarrow \neg Q, Q \Rightarrow \neg P$ and $P \Leftrightarrow \neg Q, (P \wedge \neg Q) \vee (\neg P \wedge Q)$

We use $P \Rightarrow Q \equiv \neg P \vee Q$ and $P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$

**Pair 1**

| $P$ | $Q$ | $\neg P \vee \neg Q$ | $\neg Q \vee \neg P$ |
|---|---|---|---|
| F | F | T | T |
| F | T | T | T |
| T | F | T | T |
| T | T | F | F |

Thus they are equivalent due to the commutative property!

**Pair 2**

| $P$ | $Q$ | $(\neg P \vee \neg Q) \wedge (Q \vee P)$ | $(P \wedge \neg Q) \vee (\neg P \wedge Q)$ |
|---|---|---|---|
| F | F | F | F |
| F | T | T | T |

| $P$ | $Q$ | $(\neg P \vee \neg Q) \wedge (Q \vee P)$ | $(P \wedge \neg Q) \vee (\neg P \wedge Q)$ |
|---|---|---|---|
| T | F | T | T |
| T | T | F | F |

Thus these pairs are equivalent as well!

# 2

---

**1.**

$(Smoke \Rightarrow Fire) \Rightarrow (\neg Smoke \Rightarrow Fire) \longrightarrow (\neg Smoke \vee Fire) \Rightarrow (Smoke \vee Fire) \longrightarrow \neg(\neg Smoke \vee Fire) \vee (Smoke \vee Fire) \longrightarrow (Smoke \wedge \neg Fire) \vee (Smoke \vee Fire) \longrightarrow (Smoke \vee Smoke \vee Fire) \wedge (\neg Fire \vee Smoke \vee Fire) \longrightarrow (Smoke \vee Fire) \wedge (True) \longrightarrow Smoke \vee Fire$

| **Smoke** | **Fire** | $Smoke \vee Fire$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |

Thus the sentence is neither unsatisfiable nor valid.

**2.**

$(Smoke \Rightarrow Fire) \Rightarrow ((Smoke \vee Heat) \Rightarrow Fire) \longrightarrow (\neg Smoke \vee Fire) \Rightarrow (\neg(Smoke \vee Heat) \vee$

$Fire) \longrightarrow \neg(\neg Smoke \vee Fire) \vee (\neg(Smoke \vee Heat) \vee Fire) \longrightarrow (Smoke \wedge Fire) \vee ((\neg Smoke \wedge \neg Heat) \vee Fire) \longrightarrow (\neg Smoke \wedge \neg Heat) \vee Fire$

| **Smoke** | **Fire** | **Heat** | $(\neg Smoke \wedge \neg Heat) \vee Fire$ |
|---|---|---|---|
| F | F | F | T |
| F | F | T | F |
| F | T | F | T |
| F | T | T | T |
| T | F | F | F |
| T | F | T | F |
| T | T | F | T |
| T | T | T | T |

This sentence is also neither unsatisfiable nor valid.

**3.**

$((Smoke \vee Heat) \Rightarrow Fire) \Leftrightarrow ((Smoke \Rightarrow Fire) \vee (Heat \Rightarrow Fire)) \longrightarrow (\neg(Smoke \vee Heat) \vee Fire) \Leftrightarrow (\neg Smoke \vee Fire \vee \neg Heat) \longrightarrow (\neg(\neg(Smoke \vee Heat) \vee Fire) \vee (\neg Smoke \vee Fire \vee \neg Heat)) \wedge (\neg(\neg Smoke \vee Fire \vee \neg Heat) \vee (\neg(Smoke \vee Heat) \vee Fire)) \longrightarrow (Smoke \wedge Heat) \vee (\neg Smoke \wedge \neg Heat) \vee Fire$

| **Smoke** | **Heat** | **Fire** | $(Smoke \wedge Heat) \vee (\neg Smoke \wedge \neg Heat) \vee Fire$ |
|---|---|---|---|
| F | F | F | T |

| Smoke | Heat | Fire | $(Smoke \land Heat) \lor (\neg Smoke \land \neg Heat) \lor Fire$ |
|---|---|---|---|
| F | F | T | T |
| F | T | F | F |
| F | T | T | T |
| T | F | F | F |
| T | F | T | T |
| T | T | F | T |
| T | T | T | T |

This sentence is neither unsatisfiable nor valid.

# 3

### a)

- Mythical: $A$
- Mortal: $B$
- Mammal: $C$
- Horned: $D$
- Magical: $E$

$$A \Rightarrow \neg B, \neg A \Rightarrow B \land C, (\neg B \lor C) \Rightarrow D, D \Rightarrow E$$

$\neg A \lor \neg B, A \lor (B \land C), (B \land \neg C) \lor D, \neg D \lor E$

$(\neg A \lor \neg B) \land (A \lor B) \land (A \lor C) \land ((B \land \neg C) \lor D) \land (\neg D \lor E)$

**b)**

$(\neg A \lor \neg B) \land (A \lor B) \land (A \lor C) \land (B \lor D) \land (\neg C \lor D) \land (\neg D \lor E)$

**c)**

Examining the first two clauses of this knowledge base we see $(\neg A \lor \neg B) \land (A \lor B)$ which is an XOR gate meaning a unicorn is mythical only when it is immortal, and if it is immortal it is not mythical. The unicorn can be proven to be magical if it is horned. It is horned if it is immortal or a mammal.

# 4

---

- Figure 1
  - Decomposable? Yes: all `AND` gates are fed clauses of independent variables that don't overlap
  - Deterministic? No: All **CHILD** `OR` gates are deterministic, however the **root** `OR` node **CAN** have both branches of the `OR` evaluate to true thus it is not deterministic overall.
  - Smooth? No: Some of the child `OR` s have inputs of different variables, e.g. one `OR` gate has input $C$ and then $\neg C \land \neg D$
- Figure 2
  - Decomposable? Yes: All `AND` gates are fed clauses of independent variables that don't overlap
  - Deterministic? No: Not every `OR` node has children that are inconsistent
  - Smooth? Yes: Every `OR` gate has children that are made up of the same set of variables

# 5

### a)

| $A$ | $B$ | $(\neg A \land B) \lor (\neg B \land A)$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |

$$WMC = \omega(A, \neg B) + \omega(\neg A, B) = 0.1 \cdot 0.7 + 0.9 \cdot 0.3 = 0.34$$

### b)

The count here is the same if we treat the `AND` nodes as products of their children and `OR` nodes as sums. This just gives us the same calculation we did above due to the helpful properties of DNF. More specifically, since the root node is an `OR` we know the circuit is truth when either of the child `AND` nodes are true. They cannot both be true since the circuit is deterministic and thus we can simply add the weights of both when true and this gives us the total weight for the circuit when satisfied.

### c)

Here we will compute the $WMC$ using the same strategy as above: `OR` s will use addition of child nodes and `AND` s will use multiplication of child nodes.

## Bottom `AND` Layer

$0.9 \cdot 0.3, 0.9 \cdot 0.7, 0.3 \cdot 0.1, 0.7 \cdot 0.1, 0.5 \cdot 0.3, 0.5 \cdot 0.7, 0.3 \cdot 0.5, 0.7 \cdot 0.5$

## Bottom OR Layer

$0.27 + 0.07, 0.35 + 0.15, 0.63 + 0.03, 0.15 + 0.35$

## Top AND Layer

$0.34 \cdot 0.5, 0.66 \cdot 0.5$

## Top OR Layer

$0.17 + 0.33$

## Entire Circuit

$WMC = 0.5$