AI-Assignment-07

Manoj Kolpe Lingappa

1. a. Enumeration
   It is the process of listing all possible models for the given environment.
   b. Validity
   A sentence is said to be valid if it is true in all the models. Eg. P v¬ P is always True, hence that sentence is said to be valid.
   c. Satisfiability
   A sentence is said to be satisfiable if it is true in some model. Suppose R1,R2,R3 be some model in which R1 and R2 are true, then R1∧R2∧R3 is said to be satisfiable. A sentence is said to be unsatisfiable if it's false in all the model.
   d. CNF and DNF
   A sentence expressed as a conjunction of disjunctions of literals are called as the Conjuctive Normal Form.
   Ex. (A v B v C) ∧ (C v D), where A, B, C, D are literals.
   A sentence expressed as a disjunction of conjunction of literals are called as the Disjunctive Normal Form.
   Ex. (A ∧ B ∧ C) v (C ∧ D), where A, B, C, D are called literals.
   e. Resolution
   One way of proving things in proportional logic is means of resolution. We can take a sentence that is true and generate new sentence that is also true. There is a resolution rule say's
   P v Q v ¬ R.

2.
a. [(Food ⇒Party) v (Drinks⇒Party)] ⇒ [(Food ^ Drinks) ⇒Party] [1]

m = (Food ^ Drinks) ⇒Party
n = (Food ⇒Party) v (Drinks⇒Party)
o = [(Food ⇒Party) v (Drinks⇒Party)] ⇒ [(Food ^ Drinks) ⇒Party]

Food	Party	Drinks	Food ⇒ Party	Drinks⇒Party	Food ^ Drinks	m	n	o
F	F	F	T	T	F	T	T	T
F	F	T	T	F	F	T	T	T
F	T	F	F	F	F	F	F	T
F	T	T	F	T	F	F	F	T
T	F	F	F	T	F	T	T	T
T	F	T	F	F	T	F	F	T
T	T	F	T	F	F	F	F	T
T	T	T	T	T	T	T	T	T

¬[(Food ⇒Party) v (Drinks⇒Party)] v [(Food ^ Drinks) ⇒Party]
¬[¬Food v Party v ¬Drinks v Party] v [¬(Food ^ Drinks) v Party]
¬[¬Food v Party v ¬Drinks v Party] v [¬Food v ¬Drinks v Party]
[Food ∧ ¬Party ∧ Drinks ∧ ¬Party] v [¬Food v ¬Drinks v Party]
(Food v [¬Food v ¬Drinks v Party]) ∧ (¬Party v [¬Food v ¬Drinks v Party]) ∧ (Drinks v [¬Food v ¬Drinks v Party]) ∧ (¬Party v [¬Food v ¬Drinks v Party])

(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party) ∧ (Drinks v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party)
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party) ∧ (Drinks v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party)
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v Party v ¬Food v ¬Drinks) ∧ (Drinks v ¬Drinks v ¬Food v Party) ∧ (¬Party v Party v ¬Food v ¬Drinks)
(True) ∧ (True) ∧ (True) ∧ (True)
True. Hence the given sentence is **valid**.
The above sentence is valid since it's **TRUE** in all the models.
The above sentence is satisfiable since it's **TRUE** in all models.
b.
1. left-hand
(Food ⇒ Party) v (Drinks ⇒ Party)
(¬Food v Party) v (¬Drinks v Party)
¬Food v ¬Drinks v Party

2. right-hand
(Food ^ Drinks) ⇒ Party
¬(Food ^ Drinks) v Party
¬Food v ¬Drinks v Party
Both sides of the implication are equal, which means that no matter the model they will always be the same (true in all models). Hence, the sentence is valid.

¬[(Food ⇒Party) v (Drinks⇒Party)] v [(Food ^ Drinks) ⇒Party]
¬[¬Food v Party v ¬Drinks v Party] v [¬(Food ^ Drinks) v Party]
¬[¬Food v Party v ¬Drinks v Party] v [¬Food v ¬Drinks v Party]
[Food ∧ ¬Party ∧ Drinks ∧ ¬Party] v [¬Food v ¬Drinks v Party]
(Food v [¬Food v ¬Drinks v Party]) ∧ (¬Party v [¬Food v ¬Drinks v Party]) ∧ (Drinks v [¬Food v ¬Drinks v Party]) ∧ (¬Party v [¬Food v ¬Drinks v Party])
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party) ∧ (Drinks v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party)
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party) ∧ (Drinks v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party)
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v Party v ¬Food v ¬Drinks) ∧ (Drinks v ¬Drinks v ¬Food v Party) ∧ (¬Party v Party v ¬Food v ¬Drinks)
c.
To prove this by resolution, we need to prove that (LHS ^ ¬RHS) is unsatisfiable. We can take LHS and RHS from (b)
(¬Food v ¬Drinks v Party) ^ ¬(¬Food v ¬Drinks v Party)
applying De Morgan's rule
(¬Food v ¬Drinks v Party) ^ Food ^ Drinks ^ ¬Party
Then we apply the resolution rule until we obtain the empty set. The first and the second clause:
¬Food v ¬Drinks v Party   Food
¬Drinks v Party
We apply the resolution rule
¬Drinks v Party   Drinks
Party
Resolution rule with the inferred clause and the fourth clause of the original sentence
Party   ¬Party
{}
Hence empty set and this proves that the sentence is valid.

¬[(Food ⇒Party) v (Drinks⇒Party)] v [(Food ^ Drinks) ⇒Party]
¬[¬Food v Party v ¬Drinks v Party] v [¬(Food ^ Drinks) v Party]
¬[¬Food v Party v ¬Drinks v Party] v [¬Food v ¬Drinks v Party]
[Food ∧ ¬Party ∧ Drinks ∧ ¬Party] v [¬Food v ¬Drinks v Party]
(Food v [¬Food v ¬Drinks v Party]) ∧ (¬Party v [¬Food v ¬Drinks v Party]) ∧ (Drinks v [¬Food v ¬Drinks v Party]) ∧ (¬Party v [¬Food v ¬Drinks v Party])
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party) ∧ (Drinks v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party)
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party) ∧ (Drinks v ¬Food v ¬Drinks v Party) ∧ (¬Party v ¬Food v ¬Drinks v Party)
(Food v ¬Food v ¬Drinks v Party) ∧ (¬Party v Party v ¬Food v ¬Drinks) ∧ (Drinks v ¬Drinks v ¬Food v Party) ∧ (¬Party v Party v ¬Food v ¬Drinks)
(True) ∧ (True) ∧ (True) ∧ (True)
3.

A	B	C	A ∧ B	A ∧ B → C	A → C	B → C	(A → C) v (B → C)
F	F	F	F	T	T	T	T
F	F	T	F	F	F	F	F
F	T	F	F	T	T	F	T
F	T	T	F	F	F	T	T
T	F	F	F	T	F	T	T
T	F	T	F	F	T	F	T
T	T	F	T	F	F	F	F
T	T	T	T	T	T	T	T

(A → C) v (B → C)
(¬A v C) v (¬B v C)
(¬A v ¬B v C)
(¬(A ^ B) v C)
we know (A ^ B) → C, hence
(¬C v C)
True.

4.
a. ¬P ^ ¬Q ⟺ ¬(P v Q)
(¬P ^ ¬Q ⟹ ¬(P v Q)) ∧ (¬(P v Q) ⟹ (¬P ^ ¬Q))
(¬(¬P ^ ¬Q) v (¬P ^ ¬Q)) ∧ ((P v Q) v (¬P ^ ¬Q))
((P v Q) v (¬P ^ ¬Q))∧((P v Q) v (¬P ^ ¬Q))
(((P v Q) v ¬P)∧(¬Q v (P v Q))) v (((P v Q) v ¬P) ^ ((P v Q) v ¬Q))
((True) ∧ (True)) v (((P v Q) v ¬P) ^ ((P v Q) v ¬Q))
True

b. ¬(P∧Q) ⟺¬P v ¬Q
(¬(P∧Q) ⟹ (¬ P v ¬Q)) ∧ ((¬ P v ¬Q) ⟹ ¬(P∧Q))
((¬P v ¬Q) ⟹ (¬ P v ¬Q)) ∧ ((¬ P v ¬Q) ⟹ ¬(P∧Q))
(¬(¬P v¬Q) v (¬ P v ¬Q)) ∧ (¬(¬ P v ¬Q) v (¬P v ¬Q))
((P ∧ Q) v (¬ P v ¬Q)) ∧ ((P ∧ Q) v (¬P v ¬Q))
(((¬ P v ¬Q) v P) ∧ ((¬ P v ¬Q) v Q)) ∧ (((¬P v ¬Q) v P) ∧ ((¬P v ¬Q) v Q))
True ∧ True ∧ True ∧ True

True

c.

$(P \vee (P \wedge Q)) \Longleftrightarrow P$

$(P \vee (P \wedge Q) \Longrightarrow P) \wedge (P \Longrightarrow P \vee (P \wedge Q))$

$(\neg(P \vee (P \wedge Q)) \vee P) \wedge (\neg P \vee (P \vee (P \wedge Q)))$

$((\neg P \wedge (\neg P \vee \neg Q)) \vee P) \wedge (\neg P \vee ((P \vee P) \wedge (P \vee Q)))$

$(((\neg P \wedge \neg P) \vee (\neg P \wedge \neg Q)) \vee P)) \wedge (\neg P \vee (P \wedge (P \vee Q)))$

$((\neg P \vee (\neg P \wedge \neg Q)) \vee P) \wedge (\neg P \vee ((P \wedge P) \vee (P \wedge Q))$

$((\neg P \vee \neg P) \wedge (\neg P \vee \neg Q)) \vee P) \wedge (\neg P \vee (P \vee (P \wedge Q)))$

$(\neg P \wedge (\neg P \vee \neg Q)) \vee P) \wedge (\neg P \vee P)$

$(\neg P \vee P) \wedge (\neg P \vee P)$

True ∧ True

True

1. http://www4.di.uminho.pt/~jba/tmp/slides-ValidityChecking-handout.pdf
2. http://intrologic.stanford.edu/notes/chapter_05.html