## Question 1 - Terms

Define the following:

**1.1. Logical Entailment (4)**

- $\alpha \vDash \beta$ (1) if and only if (1), in every model in which $\alpha$ is true (1), $\beta$ is also true (1)

**1.2. Logical entailment in terms of validity and satisfiability**

- For any sentences $\alpha$ and $\beta$ (✓), $\alpha \vDash \beta$ (✓) if and only if (✓) the sentence ($\alpha \land \neg\beta$) (✓) is unsatisfiable (✓)

**1.3. Monotonicity (3)**

- The set of entailed sentences (1) can only increase (1) as information is added to the knowledge base (1). Or, for any sentences $\alpha$ and $\beta$ (1), if KB $\vDash \alpha$ (1) then KB $\land\ \beta \vDash \alpha$ (1)

**1.4. Deduction Theorem (3)**

- For any sentences $\alpha$ and $\beta$ (1), $\alpha \vDash \beta$ (1) if and only if (1) the sentence ($\alpha \Rightarrow \beta$) (1) is valid (1)

**1.5. Satisfiability (3)**

- A sentence is satisfiable (1) if it is true in, or satisfied by (1), some model (1)

**1.6. Definitive Clause (4)**

- A definite clause is a disjunction (1) of literals (1) of which exactly one (1) is positive (1)

**1.7. A sound inference algorithm**

- An inference algorithm (1) that derives (1) only entailed sentences (1) is called sound.

**1.8. A complete inference algorithm**

- An inference algorithm is complete (1) if it can derive any sentence (1) that is entailed (1)

**1.9. A proof by a contradiction**

- For any sentences $\alpha$ and $\beta$ (1), $\alpha \vDash \beta$ (1) if and only if the sentence ($\alpha \land \neg\beta$) (1) is unsatisfiable (1)

**1.10. Logical Equivalence Rules**

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{De Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

## Question 2 – Resolution

**2.1. Give the full resolution rule (5)**

$$\frac{l_i \vee \cdots \vee l_k, m_1 \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

Where $l_i$ and $m_j$ are complementary literals

**2.2. Give all the steps of the resolution algorithm (10)**

To show that KB ⊨ α, we show that KB ∧ ¬α is unsatisfiable.

- Step 1. Convert KB ∧ ¬α to conjunctive normal form (CNF).
- Step 2. The resolution rule is repeatedly applied to the resulting clauses.
- Step 3. One of two things happens:
  - There are no new clauses that can be added, in which case KB does not entail α.
  - Two clauses resolve to yield the empty clause, in which case KB entails α.

## Question 3 – Conjunctive Normal Form (CNF)

**3.1. Give the four steps to convert any logical expression in Propositional Logic into Conjunctive normal form (CNF).**

- Step 1. Eliminate ⇔, replacing α ⇔ β with (α ⇒ β) ∧ (β ⇒ α)
- Step 2. Eliminate ⇒, replacing α ⇒ β with ¬α ∨ β
- Step 3. CNF requires ¬ to appear only in literals, so we "move ¬ inwards" by repeated application of the following equivalences
  - ¬(¬α) ≡ α (double-negation elimination)
  - ¬(α ∧ β) ≡ (¬α ∨ ¬β) (De Morgan)
  - ¬(α ∨ β) ≡ (¬α ∧ ¬β) (De Morgan
- Step 4. Now we have a sentence containing nested ∧ and ∨ operators applied to literals.

## Question 4 - Contradiction

**4.1. How is a proof by contradiction performed? Answer as comprehensive as possible. (10)**

- We know that $\alpha \models \beta$ if and only if the sentence $(\alpha \wedge \neg\beta)$ is unsatisfiable. A sentence in satisfiable if it is true in, or satisfied by, some model. Satisfiability can be checked by enumerating the possible models until one is found that satisfies the sentence. Proving $\beta$ from $\alpha$ by checking the unsatisfiability of $(\alpha \wedge \neg\beta)$ corresponds exactly to the standard mathematical proof technique of reductio ad absurdum (literally, “reduction to an absurd thing”). It is also called proof by refutation or proof by contradiction. One assumes a sentence $\beta$ to be false and shows that this leads to a contradiction with known axioms $\alpha$. This contradiction is exactly what is meant by saying that the sentence $(\alpha \wedge \neg\beta)$ is unsatisfiable.

# Question 5 – Logic

**5.1. Define Logical Entailment: (3)**

- $\alpha \models \beta \Leftrightarrow M(\alpha) \subseteq M(\beta)$ *or*
- $\alpha \models \beta \Leftrightarrow$ the sentence $(\alpha \Rightarrow \beta)$ is valid *or*
- $\alpha \models \beta \Leftrightarrow$ if the sentence $(\alpha \wedge \beta)$ is unsatisfiable

**5.2. Choose the correct answers:**

- $P \vee P$ is equivalent to: (2) – a,b,c,d is correct
  - a. $P$
  - b. $P \vee P$
  - c. $\neg\neg P$
  - d. $P \wedge P$
  - e. $\neg P \wedge P$
- The following formula $(P \vee \neg Q) \wedge (\neg P \vee \neg Q)$ is: (2) – b is correct
  - a. in terme vorm / in term form
  - b. in disjunkte normaalvorm / in disjunctive normal form
  - c. in konjunkte normaalvorm / in conjunctive normal form
  - d. in atomiese vorm / in literal form
  - e. in predikaat vorm / in predicate form

**5.3. *Name and describe the five logical connectives that is used in Propositional logic to construct complex sentences:***

- $\neg$(not) (1). A sentence such as $\neg W_{1,3}$ is called the negation of $\neg W_{1,3}$ (1).
- $\wedge$(and) (1). A sentence whose main connective is $\wedge$, such as $W_{1,3} \wedge P_{1,3}$, is called a conjunction (1); its parts are the conjuncts.
- $\vee$(or) (1). A sentence using $\vee$, such as $(W_{1,3} \wedge P_{1,3}) \vee W_{2,2}$, is a disjunction (1) of the disjuncts $(W_{1,3} \wedge P_{1,3})$ and $W_{2,2}$.
- $\Rightarrow$(implies) (1). A sentence such as $(W_{1,3} \wedge P_{1,3}) \Rightarrow \neg W_{2,2}$ is called an implication (1) (or conditional). Its premise or antecedent is $(W_{1,3} \wedge P_{1,3})$, and its conclusion or consequent is $\neg W_{2,2}$.
- $\Leftrightarrow$ (if and only if) (1). The sentence $(W_{1,3} \wedge P_{1,3}) \Leftrightarrow \neg W_{2,2}$ is a biconditional (1).