|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p | q | s | |  | | --- | | p→s | | |  | | --- | | q∧(p→s) | | |  | | --- | | q∧s | | p, q ∧ (p → s) |= q ∧ s |
| T | T | T | T | T | T | T |
| T | T | F | T | F | F | F |
| T | F | T | T | F | F | F |
| T | F | F | F | F | F | F |
| F | T | T | T | T | T | F |
| F | T | F | T | T | F | F |
| F | F | T | T | F | F | F |
| F | F | F | T | F | F | F |

Logic Assessment

1.

p, q ∧ (p → s) |= q ∧ s construct a truth table to evaluate whether p and q∧(p→s) logically entail (∣=) q∧s.

From the truth table we observe that p and q∧(p→s) do not logically entail (∣=) q∧s.

2.

R = in shape

Q = healthy

P = unhealthy

R → Q |= ¬Q → (P ∨ ¬R)

If a person is in shape (R) then they are healthy (Q) and this logically entails that if a person is not healthy (¬Q) then they are unhealthy or not in shape (P ∨ ¬R) .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P | Q | R | ¬Q | R → Q | ¬Q → (P ∨ ¬R) | R → Q |= ¬Q → (P ∨ ¬R) |
| T | T | T | F | T | T | T |
| T | T | F | F | T | T | T |
| T | F | T | T | F | T | F |
| T | F | F | T | F | F | F |
| F | T | T | F | T | T | T |
| F | T | F | F | T | T | T |
| F | F | T | T | T | T | T |
| F | F | F | T | T | T | T |

Thus, the truth table shows that whenever R → Q is true, ¬Q → (P ∨ ¬R) must be true, and this demonstrates that the implication R → Q |= ¬Q → (P ∨ ¬R) holds.

3. Prove the following, using the semantic equivalences given in Section 3 of the Logic Notes. Prove step by step, using one equivalence at a time.

(a) ¬p → ¬p ≡ true  [8 marks]

LHS

¬p → ¬p ≡ ¬(¬p) ∨ p (implication)

¬(¬p) ∨ ¬p ≡ p ∨ ¬p (double negation)

p ∨ ¬p ≡ true (negation)

(b) (q → r) ∧ (p ∨ r) ≡ (p → q) → r  [16 marks]

LHS

(q → r) ∧ (p ∨ r)  ≡  ((¬q)  ∨  r)   ∧  (p ∨ r)  (implication)

((¬q)  ∨  r)   ∧  (p ∨ r)  ≡ r  ∨ ( ¬q  ∧ p ) (distributivity)

r  ∨ ( ¬q  ∧ p )  ≡ r ∨ ¬(¬p ∧ q) (de Morgan)

r ∨ ¬(¬p ∧ q) ≡ r ∨ ¬(p → q) (implication)

r ∨ ¬(p → q) ≡ ¬(p → q) ∨ r (communicativity)

¬(p → q) ∨ r ≡ (p → q) → r (implication)

4. Construct natural deduction proofs in Coq for the following formulae. Only use the natural deduction rules described in the lectures and the Logic Notes, not any other rules of Coq. The last two proofs require the classical axiom of the excluded middle. So for these you have to include the line

destruct (classic p) as [hp|hnp].

Rules: intro, apply, assumption, elimination rule, split

|  |  |
| --- | --- |
| Expression | Coq proof |
| p ∨ p → p [4 marks] | Section proof.  Variables p: Prop.  Lemma l : p \/ p -> p.  Proof.  intro hp.  destruct hp as [hp1 | hp2].  assumption.  assumption.  Qed.  End proof. |
| p → p ∧ p [4 marks] | Section proof.  Variables p: Prop.  Lemma l : p -> p /\ p.  intro hp.  split.  assumption.  assumption.  End proof. |
| p ∧ ¬p → false [4 marks] | Section proof.  Variables p false: Prop.  Lemma l : p /\ ~p -> false.  intro hp.  destruct hp as [Hp Hnp].  destruct Hnp.  assumption.  End proof. |
| (d) false → true ∧ false [4 marks] | Section proof.  Lemma l : False -> True /\ False.  intros hf.  split.  exact I.  assumption.  End proof. |
| (e) (p → q) → (¬p ∨ q) [6 marks] | Section proof.  Require Import Coq.Logic.Classical  Variables p q: Prop.  Lemma l : (p -> q) -> (~p \/ q).  intros hpq.  destruct (classic p) as [hp|nhp].  right.  apply hpq.  assumption.  left.  assumption.  End proof. |
| (f) ((p → false) → p) → p | Section proof.  Require Import Coq.Logic.Classical.  Variables p : Prop.  Lemma l: ((p -> False) -> p) -> p.  intros h.  destruct (classic p) as [hp|hnp].  exact hp.  apply h.  intro h1.  destruct hnp.  assumption.  End Proof. |

5. Translate the following English language sentences into predicate formulas, using the three predicate symbols Dragon, ParentOf and Happy. (

a) The parents of a happy dragon are happy. [8 marks]

∃d. ∀p.Dragon(d) /\ Happy(d) -> ParentOf(p) -> Happy(p).

(b) A dragon is unhappy if one of its parents is unhappy. [8 marks]

 ∃d. ∃p. ParentOf(p) /\ ¬Happy(p) -> Dragon(d) -> ¬Happy(d).

6. State for each formula whether it is unsatisfiable, satisfiable but not a tautology, or a tautology, without any explanation or proof.

1. (∀x. P(x)) ∨ (∃x. P(x)) [2 marks]

Tautology

1. (∀x. P(x)) ∧ (∃x. ¬P(x)) [2 marks]

Unsatisfiable

1. ∀x. Q(x) → P(x) ∨ Q(x) [2 marks]

Tautology

1. ∀x. P(x) ∨ Q(x) → Q(x) [2 marks]

Satisfiable but not a tautology