Exercises for CS2013

Logic Practical 2

Nov 13, 2013

Introduction

This practical covers a range of aspects of Predicate Logic.

Do as much as you can during the session, and what remains do as homework.

If you work best on your own, fine. But, if you work better with others, form peer groups. See to it that you individually know what to do and how, for at the "end of the day" the exam is you and the exam.

Predicate Logic

Vacuous Quantification

1. Formally evaluate the following expression with respect to model **M1** = (**D**,**I**) where **D**={bill, jill, mary} and **I**(jill') = jill, **I**(mary') = mary, **I**(bill') = bill, **I**(B)={bill, jill}, **I**(G)={mary}, **I**(A)={(bill, jill) , (mary, bill)}. Show the set of expressions of the form $\forall x\, [B(bill') \rightarrow A(bill', jill')](x := bill)$, $\forall x\, [B(bill') \rightarrow A(bill', jill')](x := jill)$, $\forall x\, [B(bill') \rightarrow A(bill', jill')](x := mary)$.

    $$\forall x\, [B(bill') \rightarrow A(bill',jill')]$$

Answer:
$\forall x\, [B(bill') \rightarrow A(bill',jill')]$ is T in **M1** iff for every a, x := a, where **I**(a) $\in$ D, $B(bill') \rightarrow A(bill',jill')$ is T in **M1**.

- where (x := bill'), $B(bill') \rightarrow A(bill',jill')$ is $B(bill') \rightarrow A(bill',jill')$. $B(bill') \rightarrow A(bill',jill')$ is T in M1 iff B(bill') is F in M1 or A(bill',jill') is T in M1. B(bill') is F in M1 iff $I(bill') \notin I(B)$, i.e. bill $\notin$ {bill, jill}, which is F. A(bill',jill') is T in M1 iff $(I(bill'),I(jill')) \in I(A)$, i.e. (bill, jill) $\in$ {(bill, jill) , (mary, bill)}, which is T. So, $B(bill') \rightarrow A(bill',jill')$ is T.
- where (x := jill'), $B(bill') \rightarrow A(bill',jill')$ is $B(bill') \rightarrow A(bill',jill')$. $B(bill') \rightarrow A(bill',jill')$ is T in M1 iff B(bill') is F in M1 or A(bill',jill') is T in M1. B(bill') is F in M1 iff $I(bill') \notin I(B)$, i.e. bill $\notin$ {bill, jill}, which is F. A(bill',jill') is T in M1 iff $(I(bill'),I(jill')) \in I(A)$, i.e. (bill, jill) $\in$ {(bill, jill) , (mary, bill)}, which is T. So, $B(bill') \rightarrow A(bill',jill')$ is T.
- where (x := mary'), $B(bill') \rightarrow A(bill',jill')$ is $B(bill') \rightarrow A(bill',jill')$. $B(bill') \rightarrow A(bill',jill')$ is T in M1 iff B(bill') is F in M1 or A(bill',jill') is T in M1. B(bill') is F in M1 iff $I(bill') \notin I(B)$, i.e. bill $\notin$ {bill, jill}, which is F. A(bill',jill') is T in M1 iff $(I(bill'),I(jill')) \in I(A)$, i.e. (bill, jill) $\in$ {(bill, jill) , (mary, bill)}, which is T. So, $B(bill') \rightarrow A(bill',jill')$ is T.

We see that for every possible value of x with respect to D, the truth value of ∀x [B(bill') → A(bill',jill')] is T. Note that the substitutions of values for x in ∀x [B(bill') → A(bill',jill')] does not make a difference, showing the impact of vacuous quantification.

Empty Domain

2. Formally evaluate the following expression with respect to model **M2** = (**D**,**I**) where **D**={ }, **I**(B)={}, **I**(A)={}.

   ∀x [B(x) → A(x,x)]

Answer:
∀x [B(x) → A(x,x)] is T in **M2** iff for every a, x := a, where **I**(a) ∈ D, B(a) → A(a,a) is T in **M2**. Or, in other words, ∀x [B(x) → A(x,x)] is F in **M2** iff at least one expression a substituted for x in the formula, (x := a), makes the formula [B(a) → A(a,a)] false, and true otherwise. Since there is nothing we can substitute for x, there is nothing to make the formula false. So, it is true.

False Antecedent

3. Formally evaluate the following expression with respect to model **M3** = (**D**,**I**) where **D**={bill, jill, mary} and **I**(jill') = jill, **I**(mary') = mary, **I**(bill') = bill, **I**(B)={}, **I**(G)={mary}, **I**(A)={(bill, jill) , (mary,bill)}.

   ∀x [B(x) → A(x,x)]

Answer:
∀x [B(x) → A(x,x)] is T in M1 iff for every a, x := a, **I**(a) ∈ D, B(a) → A(a,a) is T in **M3**.

- where (x := bill'), B(x) → A(x,x) is B(bill') → A(bill',bill'). B(bill') → A(bill',bill') is T in **M3** iff B(bill') is F in **M3** or A(bill',bill') is T in **M3**. B(bill') is F in M1 iff I(bill') ∉ I(B), i.e. bill ∉ {}, which is T. So, B(bill') → A(bill',bill') is T.
- where (x := jill'), B(x) → A(x,x) is B(jill') → A(jill',jill'). B(jill') → A(jill',jill') is T in **M3** iff B(jill') is F in **M3** or A(jill',jill') is T in **M3**. B(jill') is F in M1 iff I(jill') ∉ I(B), i.e. jill ∉ {}, which is T. So, B(jill') → A(jill',jill') is T.
- where (x := mary'), B(x) → A(x,x) is B(mary') → A(mary', mary'). B(mary') → A(mary', mary') is T in **M3** iff B(mary') is F in **M3** or A(mary', mary') is T in **M3**. B(mary') is F in M1 iff I(mary') ∉ I(B), i.e. mary ∉ {}, which is T. So, B(mary') → A(mary', mary') is T.

We see that in each case, the antecedent of the conditional is F, so the whole conditional is true.

Create a model

4. Provide a model **M4** = (**D**,**I**) where **D** has 6 elements, proper names have their usual interpretations, the interpretation of B, **I**(B), has at least three elements of **D**, and the interpretation of A, **I**(A), has three pairs from **D**. The model should be such that the following expressions have the indicated truth values.

   a. $\exists x \, \exists y \, \mathbf{(}B(x) \wedge A(x,y)\mathbf{)}$ **true**
   b. $\forall x \, \exists y \, \mathbf{(}B(x) \wedge A(x,y)\mathbf{)}$ **false**
   c. $\forall x \, \exists y \, \mathbf{(}B(x) \rightarrow A(x,y)\mathbf{)}$ **false**
   d. $\exists x \, \exists y \, \mathbf{(}B(x) \rightarrow A(x,y)\mathbf{)}$ **true**

Answer:
M4 = (**D**,**I**) , where **D** = {bill, jill, phil, mary, will, beth}, **I**(bill') = bill, etc, **I**(B) = {bill, phil, mary}, **I**(A) = {(mary,phil), (phil,bill), (beth,will)}.
- where (x := mary') and where (y := phil), a is T.
- where (x := beth'), b is F
- where (x := bill'), c is F.
- where (x := mary') and where (y := phil'), d is T.

Formal Evaluation of Nested Quantifiers

5. Formally evaluate the following expression in **M5** = (**D**,**I**) where **D**={bill, jill, mary}, **I**(jill') = jill, **I**(mary') = mary, **I**(bill') = bill, **I**(B) ={(bill, jill), (mary,bill), (jill, jill)}, **I**(A)={(bill, bill), (mary,jill), (jill, jill)}

$$\forall x \, \exists y \, (B(x,y) \rightarrow \exists z \, A(x,z))$$

Answer:
∀x ∃*y* (B(x,y) → ∃z A(x,z)) is T in **M5** iff for every a, x := a, where **I**(a) ∈ D, ∃*y* (B(x,y) → ∃z A(x,z)) is T in **M5**.

- where (x := bill'), ∃*y* (B(x,y) → ∃z A(x,z)) is ∃*y* (B(bill',y) → ∃z A(bill',z)). ∃*y* (B(bill',y) → ∃z A(bill',z)) is T in **M5** iff for some a, y := a, where **I**(a) ∈ D, (B(bill',y) → ∃z A(bill',z)) is T in **M5**.
    - where (y := jill'), (B(bill',y) → ∃z A(bill',z)) is (B(bill',jill') → ∃z A(bill',z)). B(bill',jill') → ∃z A(bill',z) is T in **M5** iff B(bill',jill') is F in **M5** or ∃z A(bill',z) is T in **M5**. We can see that B(bill',jill') is *not* F in **M5**, so we must consider the consequent. ∃z A(bill',z) is T in **M5** iff for some a, z := a, where **I**(a) ∈ D, A(bill',z) is T in **M5**.
        - where (z := bill'), A(bill',z) is A(bill',bill'). We can see that A(bill',bill') is T in **M5**.
- where (x := jill'), ∃*y* (B(x,y) → ∃z A(x,z)) is ∃*y* (B(jill',y) → ∃z A(jill',z)). ∃*y* (B(jill',y) → ∃z A(jill',z)) is T in **M5** iff for some a, y := a, where **I**(a) ∈ D, (B(jill',y) → ∃z A(jill',z)) is T in **M5**.
    - where (y := jill'), (B(jill',y) → ∃z A(jill',z)) is (B(jill',jill') → ∃z A(jill',z)). B(jill',jill') → ∃z A(jill',z) is T in **M5** iff B(jill',jill') is F in **M5** or ∃z A(jill',z) is T in **M5**. We can see that B(jill',jill') is *not* F in **M5**, so we must consider the consequent. ∃z A(jill',z) is T in **M5** iff for some a,

z := a, where **I**(a) ∈ D, A(jill',z) is T in **M5**.

- where (z := jill'), A(jill',z) is A(jill',jill'). We can see that A(jill',jill') is T in **M5**.

- where (x := mary'), ∃*y* (B(x,y) → ∃z A(x,z)) is ∃*y* (B(mary',y) → ∃z A(mary',z)). ∃*y* (B(mary',y) → ∃z A(mary',z)) is T in **M5** iff for some a, y := a, where **I**(a) ∈ D, (B(mary',y) → ∃z A(mary',z)) is T in **M5**.

- where (y := jill'), (B(mary',y) → ∃z A(mary',z)) is (B(mary',jill') → ∃z A(mary',z)). B(mary',jill') → ∃z A(mary',z) is T in **M5** iff B(mary',jill') is F in **M5** or ∃z A(mary',z) is T in **M5**. We can see that B(mary ',jill') is *not* F in **M5**, so we must consider the consequent. ∃z A(mary',z) is T in **M5** iff for some a, z := a, where **I**(a) ∈ D, A(mary',z) is T in **M5**.

- where (z := jill'), A(mary',z) is A(mary',jill'). We can see that A(mary',jill') is T in **M5**.

So, for every value available for x and in the antecedent and consequent, there is some y in the antecedent, and there is some z in the consequent that makes the formula true. Our calculations have been made somewhat easier since we judiciously selected values of variables. A more thorough exercise would require evaluation of *all* the alternative values.

Quantifier Equivalences

6. Suppose for a given domain *D* = {a,b,c,...} that the following quantifier expansions hold:

   a. ∀*x* *P*(*x*) ⇔ *P*(a) ∧ *P*(b) ∧ *P*(c) ∧ ...
   b. ∃*x* *P*(*x*) ⇔ *P*(a) ∨ *P*(b) ∨ *P*(c) ∨ ...

Prove that the following quantifier equivalences hold using some of the propositional equivalence laws:

1. ∀*x* *P*(*x*) ⇔ ¬∃*x* ¬*P*(*x*)

2. ∃*x* *P*(*x*) ⇔ ¬∀*x* ¬*P*(*x*)

Answer:
For 1, we assume a, that ∀*x* *P*(*x*) ⇔ *P*(a) ∧ *P*(b) ∧ *P*(c) ∧ .... The law of Double Negation allows that *P*(a) ∧ *P*(b) ∧ *P*(c) ∧ ... ⇔ ¬ ¬ (*P*(a) ∧ *P*(b) ∧ *P*(c) ∧ ...). Applying one of De Morgan's Laws to the inside negation, ¬ ¬ (*P*(a) ∧ *P*(b) ∧ *P*(c) ∧ ...) ⇔ ¬ (¬*P*(a) ∨ ¬*P*(b) ∨ ¬*P*(c) ∧ ...). Given b, (¬*P*(a) ∨ ¬*P*(b) ∨ ¬*P*(c) ∧ ...) ⇔ ∃*x* ¬*P*(*x*). And there is negation on the front, so ¬ (¬*P*(a) ∨ ¬*P*(b) ∨ ¬*P*(c) ∧ ...) ⇔ ¬ ∃*x* ¬*P*(*x*). Thus, we have proven ∀*x* *P*(*x*) ⇔ ¬∃*x* ¬*P*(*x*) (we could start the proof the other direction, but the equivalences are already stated.)

For 2, we assume b, that ∃*x* *P*(*x*) ⇔ *P*(a) ∨ *P*(b) ∨ *P*(c) ∨ ... The law of Double Negation allows that *P*(a) ∨ *P*(b) ∨ *P*(c) ∨ ... ⇔ ¬ ¬ (*P*(a) ∨ *P*(b) ∨ *P*(c) ∨ ...). Applying one of De Morgan's Laws to the inside negation, ¬ ¬ (*P*(a) ∨ *P*(b) ∨ *P*(c)

$\vee \ldots) \Leftrightarrow \neg\ (\neg\ P(\mathrm{a}) \wedge \neg\ P(\mathrm{b}) \wedge \neg\ P(\mathrm{c}) \wedge \ldots) \Leftrightarrow \neg \forall x\ \neg P(x)$. So, we have proven that $\exists x\ P(x) \Leftrightarrow \neg \forall x\ \neg P(x)$.

Quantifier Definition

7. Define a quantifier in Predicate Logic for the following two quantifiers:

    1. At least three

    2. Exactly three

Answer:
1. $\exists \mathrm{x}\ \exists \mathrm{y}\ \exists \mathrm{z}\ (\mathrm{P(x)} \wedge \mathrm{P(y)} \wedge \mathrm{P(z)} \wedge \mathrm{x} \neq \mathrm{y} \wedge \mathrm{x} \neq \mathrm{z} \wedge \mathrm{y} \neq \mathrm{z})$
2. $\exists \mathrm{x}\ \exists \mathrm{y}\ \exists \mathrm{z}\ (\mathrm{P(x)} \wedge \mathrm{P(y)} \wedge \mathrm{P(z)} \wedge \mathrm{x} \neq \mathrm{y} \wedge \mathrm{x} \neq \mathrm{z} \wedge \mathrm{y} \neq \mathrm{z} \wedge \forall \mathrm{v}\ (\mathrm{P(v)} \rightarrow (\mathrm{v} = \mathrm{x} \vee \mathrm{v} = \mathrm{y} \vee \mathrm{v} = \mathrm{z})))$