Solutions to CS511 Homework 11

Nicholas Ikechukwu - U71641768

November 21, 2024

Exercise 1. Open [LCS, page 164]: Do Exercise 2.4.12, parts (a) and (b) only.

Question Predicate Logic Formulas: For each of the formulas of predicate logic below, either find a model which does not satisfy it, or prove it is valid:

Solutions:

(a) $(\forall x \forall y (S(x,y) \to S(y,x))) \to (\forall x \neg S(x,x))$

Answer: Formula is not valid. Let us find a counterexample:

Let the domain be $\{a\}$ and interpret S as $S = \{(a,a)\}$.

In this model:

- $\forall x \forall y (S(x,y) \to S(y,x))$ is true because S is symmetric.
- $\forall x \neg S(x,x)$ is false because $S(a,a)$ is true.

This shows that the antecedent is true while the consequent is false, making the implication false.

(b) $\exists y((\forall x P(x)) \to P(y))$

Answer: I believe the Formula here is also valid. We can prove it by cases:

Case 1: If $\forall x P(x)$ is true, then $P(y)$ is true for any y, so the implication is true.

Case 2: If $\forall x P(x)$ is false, then there exists some element a in the domain for which $P(a)$ is false.

Choose $y = a$. Then both the antecedent and consequent of the implication are false, making the implication true.

In both cases, we can find a y that makes the formula true, so $\exists y((\forall x P(x)) \to P(y))$ is always true.

Exercise 2. Open [LCS, page 164]: Do Exercise 2.4.12, parts (c) and (d) only.

Question Predicate Logic Formulas: For each of the formulas of predicate logic below, either find a model which does not satisfy it, or prove it is valid:

Solutions:

(c) $(\forall x(P(x) \to \exists y Q(y))) \to (\forall x \exists y(P(x) \to Q(y)))$

Answer: I suggest that this formula is also valid. Let's prove it by contradiction:

Let's assume the antecedent is true and the consequent is false.

That is:

1) $\forall x(P(x) \to \exists y Q(y))$ is true

2) $\forall x \exists y(P(x) \to Q(y))$ is false

From the consequent, (2), there must be some a such that $\forall y(P(a) \to Q(y))$ is false.

We can tell that $P(a)$ is true and $\forall y \neg Q(y)$ is true.

However, from (1), we know that $P(a) \to \exists y Q(y)$ is true.

Since $P(a)$ is true, $\exists y Q(y)$ must be true.

It's clear that this contradicts $\forall y \neg Q(y)$.

Therefore, what we assumed, must be false, and the formula is valid.

(d) $(\forall x \exists y (P(x) \to Q(y))) \to (\forall x (P(x) \to \exists y Q(y)))$

Answer: This formula is valid. We can actually prove it directly:

Let's assume the antecedent is true: $\forall x \exists y (P(x) \to Q(y))$

Now, consider any arbitrary x:

1) If $P(x)$ is false, then $P(x) \to \exists y Q(y)$ is trivially true.

2) If $P(x)$ is true, then from our assumption, there exists a y such that $Q(y)$ is true.

This allows $\exists y Q(y)$ to be true, and thus $P(x) \to \exists y Q(y)$ is true.

In both cases, $P(x) \to \exists y Q(y)$ is true for any x.

Therefore, $\forall x (P(x) \to \exists y Q(y))$ is true, making the entire implication true.

Hence, the formula is valid.

PROBLEM 1. Open EML.Chapter 6.pdf : Do Exercise 100 on page 62.

Question

Let R be a unary relation symbol and consider the following inductively defined translation from PL to FOL, $\langle\rangle : \mathrm{WFF}_{\mathrm{PL}}(P) \to \mathrm{WFF}_{\mathrm{FOL}}(\{R\}, P)$:

$$\begin{aligned}
\langle p_i \rangle &\stackrel{\text{def}}{=} R(p_i) \text{ for every } p_i \in P,\\
\langle \bot \rangle &\stackrel{\text{def}}{=} \bot,\\
\langle \top \rangle &\stackrel{\text{def}}{=} \top,\\
\langle \neg\phi \rangle &\stackrel{\text{def}}{=} \neg\langle\phi\rangle,\\
\langle \phi \diamond \psi \rangle &\stackrel{\text{def}}{=} \langle\phi\rangle \diamond \langle\psi\rangle \text{ where } \diamond \in \{\wedge, \vee, \to\}.
\end{aligned}$$

Give a rigorous argument, using structural induction, to establish the following assertions:

1. ϕ is satisfiable in the sense of PL iff $\langle\phi\rangle$ is satisfiable in the sense of FOL.
2. ϕ is valid in the sense of PL iff $\langle\phi\rangle$ is valid in the sense of FOL.

Note that the assertions in parts 1 and 2 involve two implications because of "iff" and each of the two implications have to be proved separately.
Someone suggested the following translation $[\![]\!] : \mathrm{WFF}_{\mathrm{PL}}(P) \to \mathrm{WFF}_{\mathrm{FOL}}(\{R\}, P)$ instead of $\langle\rangle$:

$$[\![\phi]\!] \stackrel{\text{def}}{=} \langle\phi\rangle \wedge (\exists x R(x) \wedge (\exists x \neg R(x))$$

where x is a fresh first-order variable not in P. Although the translation $[\![]\!]$ can be used instead of $\langle\rangle$, and was presented as being "better" than $\langle\rangle$, give a rigorous argument for the following:

3. The added requirement in the translation $[\![]\!]$, expressed by $(\exists x R(x) \wedge (\exists x \neg R(x))$, is not necessary for correct proofs of the assertions in parts 1 and 2.

First-Order Logic: Interpretation of Propositional Logic

Answer

1. Satisfiability

Via structural induction we will prove by that ϕ is satisfiable in PL iff $\langle\phi\rangle$ is satisfiable in FOL.

The base cases:

- For $p_i \in P$: we know that p_i is satisfiable in PL iff there exists a valuation v such that

$v(p_i) = \text{true}$. This is equivalent to the existence of a first-order structure $\mathcal{A}$ and variable assignment s such that $\mathcal{A} \models_s R(p_i)$, which is the definition of satisfiability for $\langle p_i \rangle$ in FOL.

- $\bot$ is not satisfiable in PL and $\langle \bot \rangle = \bot$ is not satisfiable in FOL.
- $\top$ is satisfiable in PL and $\langle \top \rangle = \top$ is satisfiable in FOL.

Inductive step: Let's assume that for ϕ and ψ, the statement holds.

- $\neg\phi$ is satisfiable in PL iff ϕ is not valid in PL. By the induction hypothesis, this is equivalent to $\langle \phi \rangle$ not being valid in FOL, which is equivalent to $\neg\langle \phi \rangle = \langle \neg\phi \rangle$ being satisfiable in FOL.
- For $\diamond \in \{\wedge, \vee, \to\}$, $\phi \diamond \psi$ is satisfiable in PL iff there exists a valuation satisfying the truth table for $\diamond$. By the induction hypothesis, this is equivalent to the existence of a first-order structure and variable assignment satisfying $\langle \phi \rangle \diamond \langle \psi \rangle = \langle \phi \diamond \psi \rangle$ in FOL.

2. Validity

The proof for validity follows a similar structure to the satisfiability proof, using the fact that a formula is valid iff its negation is not satisfiable.

3. The Unnecessary Additional Requirement

I think that the added requirement $(\exists x R(x) \wedge (\exists x \neg R(x))$ in the translation $[\![]\!]$ is not necessary for correct proofs of the assertions in parts 1 and 2 for the following reasons:

- The proofs for parts 1 and 2 rely on the structural correspondence between PL formulas and their FOL translations, which is preserved in $\langle\rangle$ and not affected by the additional conjunct in $[\![]\!]$.
- This additional conjunct is always satisfiable in FOL, as it merely asserts the existence of at least one element in the domain for which R is true and at least one for which R is false.
- The satisfiability and validity of $\langle\phi\rangle$ in FOL are independent of this additional conjunct. If $\langle\phi\rangle$ is satisfiable (or valid), then $[\![\phi]\!]$ is satisfiable (or valid) with the same model extended to satisfy the additional conjunct. Conversely, if $[\![\phi]\!]$ is satisfiable (or valid), then $\langle\phi\rangle$ must also be satisfiable (or valid) in the same model.

Therefore, while $[\![]\!]$ may provide other benefits, it barely has an effect on the correctness of the proofs for satisfiability and validity equivalence between PL and FOL translations.

ON LEAN-4

Solutions in one file at: https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw11/hw11_nicholas_ikechukwu.lean

Exercise 3. From Macbeth's book:

Solutions

https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw11/hw11_nicholas_ikechukwu.lean

Exercise 4. From Macbeth's book

https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw11/hw11_nicholas_ikechukwu.lean

PROBLEM 2. From Macbeth's book

Solution

https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw11/hw11_nicholas_ikechukwu.lean