Solutions to CS511 Homework 09

Nicholas Ikechukwu - U71641768

November 14, 2024

Exercise 1. [LCS, page 159-160]: Exercise 2.2.3. Do both part (a), on page 159, and part (b), on page 160.

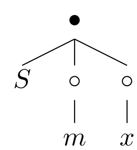
Analysis of First-Order Formulas

Part (a): Valid and Invalid Formulas in Predicate Logic

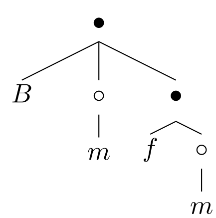
Given: m is a constant, f is a function symbol with one argument, and S and B are two predicate symbols, each with two arguments.

Valid Formulas with Parse Trees

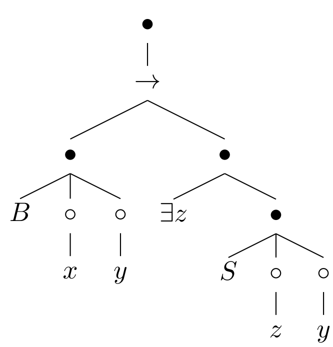
i. $S(m, x)$ is a valid formula:



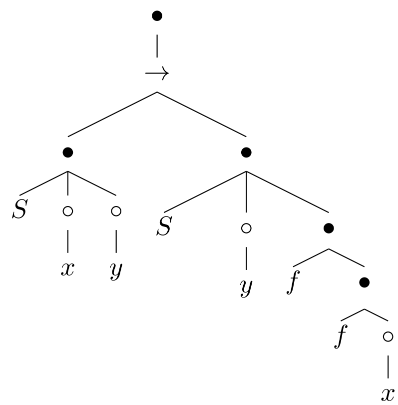
ii. $B(m, f(m))$ is a valid formula:



vi. $(B(x, y) \to (\exists z S(z, y)))$ is a valid formula:



vii. $(S(x,y) \to S(y, f(f(x))))$ is a valid formula:



Invalid Formulas with Reasons

iii. $f(m)$ is not a formula because:

- It is a term, not a formula
- No predicate symbol is applied

iv. $B(B(m,x),y)$ is not a formula because:

- B is a predicate symbol, not a function symbol
- Cannot use predicate B as argument to another predicate

v. $S(B(m),z)$ is not a formula because:

- $B(m)$ is invalid as B requires two arguments
- Cannot use predicate as argument

viii. $(B(x) \to B(B(x)))$ is not a formula because:

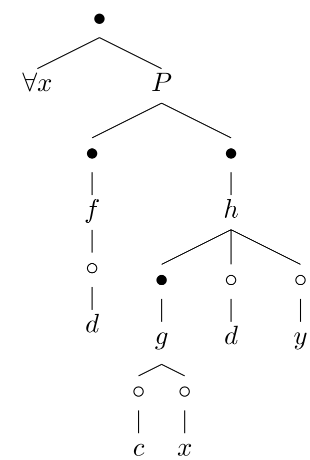
- B requires two arguments but given only one
- Cannot use predicate B as argument

Part (b)

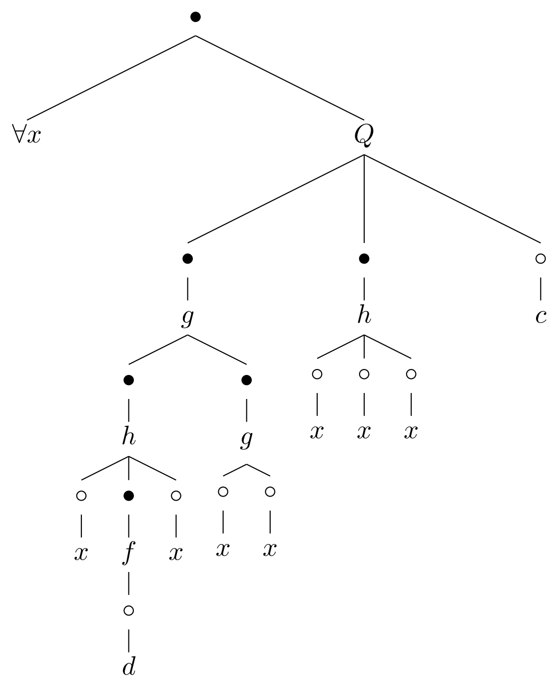
Let c and d be constants, f a function symbol with one argument, g a function symbol with two arguments, h a function symbol with three arguments, and P and Q are predicate symbols with three arguments.

Valid Formulas with Parse Trees

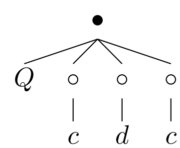
i. $\forall x P(f(d), h(g(c,x), d, y))$ is a valid formula:



iii. $\forall x Q(g(h(x, f(d), x), g(x,x)), h(x,x,x), c)$ is a valid formula:



vi. $Q(c, d, c)$ is a valid formula:



Invalid Formulas with Reasons

ii. $\forall x P(f(d), h(P(x, y), d, y))$ is not a formula because:

- Cannot use predicate P as argument to function h
- Predicates can only appear as atomic formulas

iv. $\exists z(Q(z, z, z) \to P(z))$ is not a formula because:

- P requires three arguments but given only one
- Arity mismatch for predicate P

v. $\forall x \forall y(g(x, y) \to P(x, y, x))$ is not a formula because:

- $g(x, y)$ is a term, not a formula
- Cannot use $\to$ with a term on left side
- Only atomic formulas can be connected by logical operators

Exercise 2.

Part 1: [LCS, page 160]: Exercise 2.3.2.

2.3.2. Formula Interpretation

The formula $\exists x \exists y(\neg(x = y) \wedge (\forall z((z = x) \vee (z = y))))$ specifies:

"There exist x and y, such that there is no x that equals y and for all z, z either equals x or y".

in essense:

"There exist exactly two distinct elements in the model."

This is because:

- $\exists x \exists y(\neg(x = y))$ states there are at least two different elements
- $\forall z((z = x) \vee (z = y))$ states every element must be equal to either x or y
- Together, they specify that there are exactly two distinct elements

Part 2: [LCS, page 160]: Exercise 2.3.3, modified as follows. Change part (c) to read "at least three distinct elements".

2.3.3. Predicate Logic Sentences

(a) Exactly three distinct elements:

$$\exists x \exists y \exists z(\neg(x = y) \wedge \neg(y = z) \wedge \neg(x = z) \wedge$$
$$\forall w((w = x) \vee (w = y) \vee (w = z)))$$

(b) At most three distinct elements:

$$\forall x \forall y \forall w \forall z((w = x) \vee (w = y) \vee (w = z))$$

(c) At least three distinct elements:

$$\exists x \exists y \exists z(\neg(x = y) \wedge \neg(y = z) \wedge \neg(x = z))$$

Explanations:

- For "exactly three": We state there exist three distinct elements AND every element must be one of these three
- For "at most three": We state that any fourth element must be equal to one of three elements
- For "at least three": We state there exist three elements that are all different from each other

PROBLEM 1 Open EML.Chapter 6.pdf. Do part Exercise 99 on page 61.

Interpretation of PL in FOL, II (Solution)

Let $\Sigma' \stackrel{\text{def}}{=} \{f, g_1, g_2, g_3, c_1, c_2\}$ where:

- f corresponds to $\neg$ (unary)
- g_1 corresponds to $\wedge$ (binary)
- g_2 corresponds to $\vee$ (binary)
- g_3 corresponds to $\to$ (binary)
- c_1 corresponds to $\bot$ (constant)
- c_2 corresponds to $\top$ (constant)

To define the two-element structure up to isomorphism, we construct ψ as follows:

$$
\begin{aligned}
\psi \stackrel{\text{def}}{=}\ & \exists x \exists y(\neg(x \approx y) \wedge \forall z(z \approx x \vee z \approx y))\wedge \\
& \forall x \forall y(\\
& \quad (x \approx c_1 \vee x \approx c_2)\wedge \\
& \quad \neg(c_1 \approx c_2)\wedge \\
& \quad (f(c_1) \approx c_2 \wedge f(c_2) \approx c_1)\wedge \\
& \quad (g_1(c_1, c_1) \approx c_1 \wedge g_1(c_1, c_2) \approx c_1\wedge \\
& \quad g_1(c_2, c_1) \approx c_1 \wedge g_1(c_2, c_2) \approx c_2)\wedge \\
& \quad (g_2(c_1, c_1) \approx c_1 \wedge g_2(c_1, c_2) \approx c_2\wedge \\
& \quad g_2(c_2, c_1) \approx c_2 \wedge g_2(c_2, c_2) \approx c_2)\wedge \\
& \quad (g_3(c_1, c_1) \approx c_2 \wedge g_3(c_1, c_2) \approx c_2\wedge \\
& \quad g_3(c_2, c_1) \approx c_1 \wedge g_3(c_2, c_2) \approx c_2))
\end{aligned}
$$

For any propositional formula ϕ, we construct ϕ' as:

$$\phi' \stackrel{\text{def}}{=} \phi \wedge \psi$$

Proof of Correctness:

- The sentence ψ ensures:
  - Exactly two distinct elements (c_1 and c_2 representing false and true)
  - The truth tables for negation (f), conjunction (g_1), disjunction (g_2), and implication (g_3)
  - The constants false (c_1) and true (c_2)
- If ϕ is valid in PL, then it evaluates to true under all truth assignments

- The sentence ψ ensures that any model of ϕ' is isomorphic to the two-element Boolean algebra
- Therefore, ϕ is valid in PL if and only if ϕ' is valid in FOL

ON LEAN-4

Solutions in one file at: https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw09/hw09_nicholas_ikechukwu.lean

Exercise 3. Two closely related parts, which you have to do in Lean:

Solutions

https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw09/hw09_nicholas_ikechukwu.lean

Exercise 4. From Macbeth's book

https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw09/hw09_nicholas_ikechukwu.lean

PROBLEM 2. From Macbeth's book

Solution

https://github.com/nich-ikech/CS511-hw-macbeth/blob/main/cs511HwSolutions/hw09/hw09_nicholas_ikechukwu.lean