85

Homework 2 - Predicate Logic

CMPT333N
Due on Feb 11, 2022

Predicates

Problem 1

Identify the following as constants, variables, ground atomic formulas, or non-ground atomic formulas, using the conventions of this section.

a) CMPT333 - **VARIABLE**
b) cmpt333 - **CONSTANT**
c) 333 - **CONSTANT**
d) “cmpt333” - **CONSTANT**
e) p(X,x) - **ATOMIC FORMULA** non-ground
f) p(3,4,5) - **GROUNDED FORMULA**
g) “p(3,4,5)” - **CONSTANT**

Problem 2

(csg(“CMPT333”, S, G) AND snap(S, “C.Brown”, A, P)) → answer(G) (1)

Write an expression similar to (1) for the question “What grade did L. Van Pelt get in CMPT220?” What substitution for variables did you make to demonstrate the truth of this answer? (See Example 3)

- (csg(**“CMPT220”**, S, G) AND snap(S, **“L. Van Pelt”**, A, P)) → answer(G)
  - I substituted CMP T333 for CMPT220 and “C.Brown” for “L. Van Pelt”

Problem 3

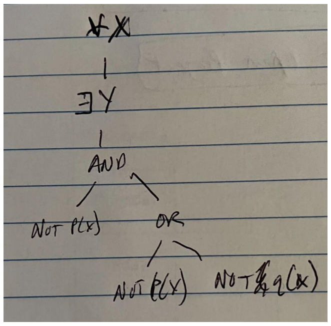
Remove redundant pairs of parentheses from the following expressions.

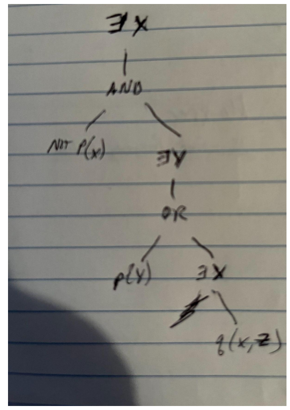
-3

- (∀X)((∃Y)(NOT(p(X) OR (p(Y) AND q(X)))))
  i) ∀X ∃Y (NOT(p(X) OR (p(Y) AND q(X))))
- (∃X)((NOTp(X)) AND ((∃Y)(p(Y)) OR (∃X)(q(X, Z))))
  i) ∃X(NOTp(X) AND ∃Y (p(Y)) OR ∃Xq(X, Z))

Problem 4

Draw expression trees for the expressions of Problem 3. Indicate for each occurrence of a variable to which quantifier, if any, it is bound.





-4

Problem 5

Rewrite the expression of Problem 3(b) so that it does not quantify the same variable twice.

- **$\exists$X$\exists$Y(NOTp(X) AND p(Y) OR q(X, Z))** X

4

Problem 6

Using the csg predicate of our running example, write expressions that assert the following.

1. C. Brown is an A student (i.e., he gets A’s in all his courses).
    a. $\forall$C(csg(C, S, G) AND snap(S, “C.Brown”, A, P)) → answer(“A”)
2. C. Brown is not an A student.
    a. $\exists$C(csg(C, S, G) AND snap(S, “C.Brown”, A, P)) → NOT answer(“A”)

✓

Problem 7

For each of the following expressions, give one interpretation that makes it true and one interpretation that makes it false.

a) ($\forall$X)($\exists$Y)(loves(X, Y))

-2 y would be 1

    i) Assumes a domain of {0, 1}
        1) **False**: *loves* is an AND function, since if x = 1 and y = 0, $\forall$1($\exists$Y)(loves(1, Y)) would be false. X
        2) **True**: *loves* is an OR function, since if x = 1 and y = 0, $\forall$1($\exists$Y)(loves(1, Y)) this would be true in all cases.

b) p(X) → NOTp(X)

i) Assumes a domain of {0, 1}

1) **True:** p(0) → NOT p(0)

2) **False**: p(1) → NOT p(1)

$P(X) \equiv True$

-2

c) $(\exists X)p(X) \rightarrow (\forall X)p(X)$

what is the interpretation of the function?

i) Assumes a domain of {0, 1}

1) **True:** If x is 0, then 0 → {0, 1} will always be true

2) **False:** If x is 1, then 1 → {0, 1} will be false when the right hand x is 0.

d) (p(X, Y) AND p(Y, Z)) → p(X, Z)

✓

i) Assumes a domain of {0, 1}

1) **True:** If p() is an AND function this will always be true

2) **False:** If p() is an OR function this breaks apart when: x = 0, y = 1, z = 0

Problem 8

Explain why each of the following are tautologies. That is, what expression(s) of predicate logic did we substitute into which tautologies of propositional logic?

a) (p(X) OR q(Y)) ≡ (q(Y) OR p(X))
    i) Commutative property, p = !p
b) (p(X, Y) AND p(X, Y)) ≡ p(X, Y)
    i) Idempotent property: (p=q) AND (p=q) → p=q
c) (p(X) → FALSE) ≡ NOT p(X)
    i) q AND !p

✓

Problem 9

Transform the following expressions into rectified expressions, that is, expressions for which no two quantifier occurrences share the same variable.

a) (∃X)(NOT p(X)) AND((∃Y)p(Y))) OR ((∃X)q(X, Z))))
    i) (∃X)(∃Y)(∃W)(NOT p(Y)) AND p(W)) OR (q(W, Z)))) ✓
b) (∃X)(∃X)p(X) OR (X)q(X) OR r(X))
    i) (∃X)((∃T)(∃R)p(T) OR q(R) OR r(X)) ✓

Problem 10

Turn the following into closed expressions by universally quantifying each of the free variables. If necessary, rename variables so that no two quantifier occurrences use the same variable.

a) p(X, Y) AND (∃Y)q(Y)
    i) (∀X)(∀R)(∃Y)p(X, Y) AND q(Y)) ✓
b) (∃X)(p(X, Y) OR (∃X)p(Y, X))
    i) (∀Y)(∃X)(∃R)(p(X, Y) OR p(Y, R))
    ii) (∀Y)(∃X)(p(X, Y) OR p(Y, X)) ✗ -2
        1) Can't rename since the X must be bounded to a single quantifier