CS 204 Discrete Mathematics

Homework 1 – Propositional Logic

Sample Solutions

1. Let the following statements be given.

p = "There is water in the cylinders."

q = "The head gasket is blown."

r = "The car will start."

(a) Translate the following statement into symbols of formal logic.

If the head gasket is blown and there's water in the cylinders, then the car won't start.

(b) Translate the following statement into everyday English.

$r \rightarrow \neg(q \vee p)$

Solution)

(a) $(q \wedge p) \rightarrow \neg r$

(b) If the car will start, then neither the head gasket is blown nor is there water in the cylinders. (or, If the car will start, then it is not the case that the head gasket is blown or there is water in the cylinders.)

3. On the basis of the following abbreviation

P: logic is enjoyable

Q: Kim will pass

R: Kim concentrates

S: the text is readable

T: Kim will secure graduation

U: Kim will be employed

V: the lectures are exciting

translate the following symbolic sentences into natural English:

(a) $P \rightarrow (Q \rightarrow R)$

(b) $(R \rightarrow Q) \rightarrow P$

(c) $S \rightarrow (P \rightarrow (\neg Q \rightarrow \neg R))$

Solution)

(a) If logic is enjoyable, the Kim will pass only if she concentrates.

(b) If Kim will pass provided that she concentrates, then logic is enjoyable.

(c) If the text is readable, then, provided that logic is enjoyable, Kim will not pass only if she does not concentrate.

5. Given the alphabet of {P, P1, ..., Q, Q1, ..., R, R1, ..., ..., ¬, ∧, ∨, →, (,) }, write a BNF grammar that generates all legal propositional formulas. You can start with the following BNF grammar rules:

digit ::= “0” | “1” | “2” | “3” | ... | “8” | “9”
integer ::= digit | digit, integer
A ::= P | P, integer // generates P, P1, ...
B ::= Q | Q, integer // generates Q, Q1, ...
C ::= R | R, integer // generates R, R1, ...

(Hint: You do not need to consider priorities of connectives. It suffices to generate fully parenthesized formulas that have no omission of parentheses.)

Solution) Propostion ::= CompoundProposition | AtomicProposition
CompoundProposition ::= Negation | Conjunction | Disjunction | Implication
AtomicProposition ::= A | B | C
Negation ::= (, ¬, Proposition,)
Conjunction ::= (, Proposition, ∧, Proposition,)
Disjunction ::= (, Proposition, ∨, Proposition,)
Implication ::= (, Proposition, →, Proposition,)