Section 03: Propositions and Proofs

1. Canonical Forms

Consider the boolean functions $F(A, B, C)$ and $G(A, B, C)$ specified by the following truth table:

A	B	C	$F(A,B,C)$	$G(A,B,C)$
1	1	1	1	0
1	1	0	1	1
1	0	1	0	0
1	0	0	0	0
0	1	1	1	1
0	1	0	1	0
0	0	1	0	1
0	0	0	1	0

(a) Write the DNF and CNF expressions for $F(A, B, C)$.

(b) Write the DNF and CNF expressions for $G(A, B, C)$.

2. Translate to Logic

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

(a) Every user has access to an electronic mailbox.

(b) The system mailbox can be accessed by everyone in the group if the file system is locked.

(c) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.

(d) At least one router is functioning normally if the throughput is between 100kbps and 500 kbps and the proxy server is not in diagnostic mode.

3. Translate to English

Translate these system specifications into English where $F(p)$ is "Printer p is out of service", $B(p)$ is "Printer p is busy", $L(j)$ is "Print job j is lost," and $Q(j)$ is "Print job j is queued". Let the domain be all printers and all print jobs.

(a) $\exists p\ (F(p) \wedge B(p)) \rightarrow \exists j\ L(j)$

(b) $(\forall j\ B(j)) \rightarrow (\exists p\ Q(p))$

(c) $\exists j\ (Q(j) \wedge L(j)) \rightarrow \exists p\ F(p)$

(d) $(\forall p\ B(p) \wedge \forall j\ Q(j)) \rightarrow \exists j\ L(j)$

4. Domain Restriction

Translate each of the following sentences into logical notation. These translations require some of our quantifier tricks. You may use the operators $+$ and $\cdot$ which take two numbers as input and evaluate to their sum or product, respectively. Remember:

- To restrict the domain under a $\forall$ quantifier, add a hypothesis to an implication.
- To restrict the domain under an $\exists$ quantifier, AND in the restriction.
- If you want variables to be different, you have to explicitly require them to be not equal.

(a) Domain: Positive integers; Predicates: `Even`, `Prime`, `Equal`
"There is only one positive integer that is prime and even."

(b) Domain: Real numbers; Predicates: `Even`, `Prime`, `Equal`
"There are two different prime numbers that sum to an even number."

(c) Domain: Real numbers; Predicates: `Even`, `Prime`, `Equal`
"The product of two distinct prime numbers is not prime."

(d) Domain: Real numbers; Predicates: `Even`, `Prime`, `Equal`, `Postivite`, `Greater`, `Integer`
"For every positive integer, there is a greater even integer"

5. Quantifier Switch

Consider the following pairs of sentences. For each pair, determine if one implies the other, if they are equivalent, or neither.

(a) $\forall x\ \forall y\ P(x, y)$ $\qquad$ $\forall y\ \forall x\ P(x, y)$

(b) $\exists x\ \exists y\ P(x, y)$ $\qquad$ $\exists y\ \exists x\ P(x, y)$

(c) $\forall x\ \exists y\ P(x, y)$ $\qquad$ $\forall y\ \exists x\ P(x, y)$

(d) $\forall x\ \exists y\ P(x, y)$ $\qquad$ $\exists x\ \forall y\ P(x, y)$

(e) $\forall x\ \exists y\ P(x, y)$ $\qquad$ $\exists y\ \forall x\ P(x, y)$

6. Quantifier Ordering

Let your domain of discourse be a set of `Element` objects given in a list called `Domain`. Imagine you have a predicate `pred`(x, y), which is encoded in the java method `public boolean pred(int x, int y)`. That is you call your predicate `pred` true if and only if the java method returns `true`.

(a) Consider the following Java method:

```
public boolean Mystery(Domain D){
    for(Element x : D) {
        for(Element y : D) {
            if(pred(x,y))
                return true;
        }
    }
}
```

`Mystery` corresponds to a quantified formula (for `D` being the domain of discourse), what is that formula?

(b) What formula does `mystery2` correspond to

```
public boolean Mystery2(Domain D){
    for(Element x : D) {
        boolean thisXPass = false;
        for(Element y : D) {
            if(pred(x,y))
                thisXPass = true;
        }
        if(!thisXPass)
            return false;
    }
    return true;
}
```

7. All for 1 and One $\forall$

Let the domain of discourse contain only the two object a and b. *For this problem only*, you are allowed to use the following <u>fake</u> equivalence rules

$$\forall x\, P(x) \equiv P(a) \wedge P(b) \qquad \forall \to \wedge$$
$$\exists x\, P(x) \equiv P(a) \vee P(b) \qquad \exists \to \vee$$

(a) Use a chain of equivalences to show that $Q \wedge (\exists x\, P(x)) \equiv \exists x\, Q \wedge P(x)$.

(b) Likewise show that $Q \vee (\exists x\, P(x)) \equiv \exists x\, Q \vee P(x)$.

(c) Are each of these equivalences also true assuming our fake equivalences? Yes or no.

i $Q \wedge (\forall x\, P(x)) \equiv \forall x\, Q \wedge P(x)$

ii $Q \vee (\forall x\, P(x)) \equiv \forall x\, Q \vee P(x)$.

(d) Do the equivalences proven in (a)-(b) hold in every other domain of discourse? <u>Briefly</u> explain why or why not.

8. Find the Bug

Each of these inference proofs is incorrect. Identify the line (or lines) which incorrectly apply a law, and explain the error. Then, if the claim is false, give concrete examples of propositions to show it is false. If it is true, write a correct proof.

(a) This proof claims to show that given $a \to (b \vee c)$, we can conclude $a \to c$.

1. $a \to (b \vee c)$ [Given]

2.1. a	[Assumption]
2.2. $\neg b$	[Assumption]
2.3. $b \vee c$	[Modus Ponens, from 1 and 2.1]
2.4. c	[$\vee$ elimination, from 2.2 and 2.3]

2. $a \to c$ [Direct Proof Rule, from 2.1-2.4]

(b) This proof claims to show that given $p \to q$ and r, we can conclude $p \to (q \vee r)$.

1.$p \to q$	[Given]
2.r	[Given]
3.$p \to (q \vee r)$	[Intro $\vee$ (1,2)]

(c) This proof claims to show that given $p \to q$ and q that we can conclude p

1.$p \to q$	[Given]
2.q	[Given]
3.$\neg p \vee q$	[Law of Implication (1)]
4.q	[Eliminate $\vee$ (2,3)