Section 02: Digital Logic and Equivalence Proofs

1. Equivalences

Prove that each of the following pairs of propositional formulae are equivalent using propositional equivalences.

(a) $p \leftrightarrow r \equiv (p \land r) \lor (\neg p \land \neg r)$ You may use the rule $p \leftrightarrow r \equiv (p \to r) \land (r \to p)$.

(b) $\neg s \to (q \to r) \equiv q \to (s \lor r)$

2. Canonical Forms

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

A	B	C	$F(A, B, C)$	$G(A, B, C)$
T	T	T	T	F
T	T	F	T	T
T	F	T	F	F
T	F	F	F	F
F	T	T	T	T
F	T	F	T	F
F	F	T	F	T
F	F	F	T	F

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

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

3. 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.

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

5. ctrl-z

Translate these logical expressions to English. For each of the translations, assume that domain restriction is being used and take that into account in your English versions.

Let your domain be all UW Students. Predicates $\texttt{143Student}(x)$ and $\texttt{311Student}(x)$ mean the student is in CSE 143 and 311, respectively. $\texttt{BioMajor}(x)$ means x is a bio major, `DidHomeworkOne(x)` means the student did homwork 1 (of 311). Finally $\texttt{KnowsJava}(x)$ and $\texttt{KnowsDeMorgan}(x)$ mean x knows Java and knows DeMorgan's Laws, respectively.

(a) $\forall x(\texttt{143Student}(x) \rightarrow \texttt{KnowsJava}(x))$

(b) $\exists x(\texttt{143Student}(x) \wedge \texttt{BioMajor}(x))$

(c) $\forall x([\texttt{311Student}(x) \wedge \texttt{DidHomeworkOne(x)}] \rightarrow \texttt{KnowsDeMorgan}(x))$

6. Domain Restriction Negated

When we negate a sentence with a domain restriction, the restriction itself remains intact (i.e. not negated) after we have fully simplified. That should make sense; if I claim something is true for every even number, I won't be convinced by you showing you an odd number. In this problem, you'll do the algebra to see why.

(a) Consider the statement $\neg\exists x\exists y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \wedge [P(x,y) \wedge Q(x,y)])$. We know that we should end up with $\forall x\forall y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \rightarrow [\neg P(x,y) \vee \neg Q(x,y)])$ (that is flip the quantifiers, rewrite the domain restriction, and negate the other requirements).

But it can help to see the full algebra written out – write out a step-by-step simplification to get the simplified form (you don't have to label with rules).

(b) Now do the same process for: $\neg\forall x\forall y(\texttt{Domain1}(x) \wedge \texttt{Domain2}(y) \rightarrow [P(x,y) \wedge Q(x,y)])$