CSE 311: Foundations of Computing I

Section 2: Logical Equivalences, Predicate Translation

1. Boolean Algebra and Digital Circuits

Consider the following propositional logic expression:

$$\neg(\neg p \vee (p \wedge \neg r))$$

(a) Translate the proposition to Boolean Algebra notation. Do not simplify the expression.

(b) Write the proposition as a Digital Circuit.

2. Logical Equivalences

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

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

(b) $\neg p \rightarrow (s \rightarrow r) \equiv s \rightarrow (p \vee r)$

(c) $\neg p \vee ((q \wedge p) \vee (\neg q \wedge p)) \equiv \mathsf{T}$

(d) $((p \wedge q) \rightarrow r) \equiv (p \rightarrow r) \vee (q \rightarrow r)$

3. DNFs and CNFs

Consider the following boolean functions $A(p, q, r)$ and $B(p, q, r)$.

p	q	r	$A(p,q,r)$	$B(p,q,r)$
T	T	T	F	T
T	T	F	F	T
T	F	T	T	T
T	F	F	F	F
F	T	T	T	F
F	T	F	T	T
F	F	T	F	T
F	F	F	F	F

(a) Write the DNF (ORs of ANDs) and CNF (ANDs of ORs) expressions for $A(p, q, r)$.

(b) Write the DNF (ORs of ANDs) and CNF (ANDs of ORs) expressions for $B(p, q, r)$.

4. Predicate Logic to English

Translate the following sentences to English. The domain of discourse is penguins. The predicates are defined as follows: Love(x, y) ::= "x loves y", Dances(x) ::= "x dances", Sings(x) ::= "x sings".

(a) $\neg\exists x(\text{Dances}(x))$

(b) $\exists x \forall y(\text{Loves}(x, y))$

(c) $\forall x(\text{Dances}(x) \to \exists y(\text{Loves}(y, x)))$

(d) $\exists x \forall y((\text{Dances}(y) \wedge \text{Sings}(y)) \to \text{Loves}(x, y))$

5. English to Predicate Logic

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 a positive integer that is prime and even."

(b) Domain: Positive integers; Predicates: Even, Prime, Equal
"If a positive integer is even, its square is also even."

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

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

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

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

6. 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) $\forall x \; \exists y \; P(x, y)$ $\qquad$ $\exists y \; \forall x \; P(x, y)$