CSE 390Z: Mathematics for Computation Workshop

Week 3 Workshop

Conceptual Review

(a) What are CNF and DNF forms?

(b) What do "tautology", "contradiction", and "contingency" mean?

(c) What is a predicate, a domain of discourse, and a quantifier?

(d) When translating to predicate logic, how do you restrict to a smaller domain in a "for all"? How do you restrict to a smaller domain in an "exists"?

1. Equivalences: Boolean Algebra

(a) Prove $p' + (p \cdot q) + (q' \cdot p) = 1$ via equivalences. Use Boolean Algebra notation.

(b) Prove $(p' + q) \cdot (q + p) = q$ via equivalences. Use Boolean Algebra notation.

2. 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

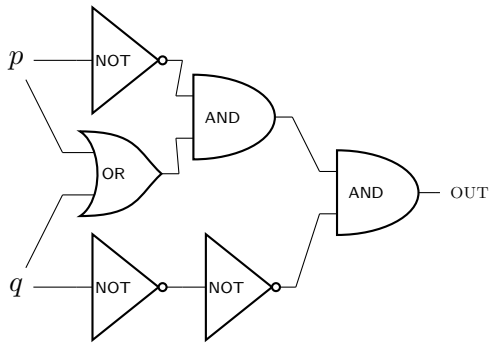
(a) Write the DNF (sum of products) and CNF (product of sums) expressions for $A(p, q, r)$.

(b) Write the DNF (sum of products) and CNF (product of sums) expressions for $B(p, q, r)$.

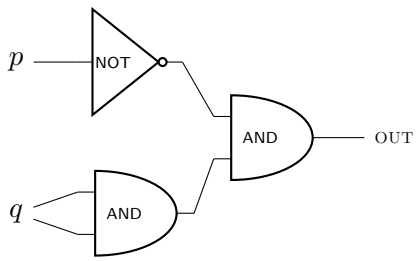
3. Circuits

Convert the following circuits into logical expressions.

(i)



(ii)



4. Predicate Logic Translation

Let the domain of discourse be all animals. Let $\text{Cat}(x) ::=$ "x is a cat" and $\text{Blue}(x) ::=$ "x is blue". Translate the following statements to English.

(a) $\forall x(\text{Cat}(x) \wedge \text{Blue}(x))$

(b) $\forall x(\text{Cat}(x) \rightarrow \text{Blue}(x))$

(c) $\exists x(\text{Cat}(x) \wedge \text{Blue}(x))$

Kabir translated the sentence "there exists a blue cat" to $\exists x(\text{Cat}(x) \rightarrow \text{Blue}(x))$. This is wrong! Let's understand why.

(d) Use the Law of Implications to rewrite Kabir's translation without the $\rightarrow$.

(e) Translate the predicate from (d) back to English. How does this differ from the intended meaning?

(f) This is a warning to be very careful when including an implication inside an exists! It should almost always be avoided, unless there is a forall involved as well. (Nothing to write for this part).

5. Domains of Discourse

For the following, find a domain of discourse where the following statement is true and another where it is false. Note that for the arithmetic symbols to make sense, the domains of discourse should be sets of numbers.

(a) $\exists x(2x = 0)$

(b) $\forall x(2x = 0)$

(c) $\exists x \exists y(x + y = 0)$

(d) $\exists x \forall y(x + y = y)$

6. English to Predicate Logic

Express the following sentences in predicate logic. The domain of discourse is penguins. You may use the following predicates: Love(x, y) ::= "x loves y", Dances(x) ::= "x dances", Sings(x) ::= "x sings".

(a) All penguins that dance cannot sing.

(b) There is a penguin that dances and sings.

(c) There is a penguin that dances, and there is a penguin that sings. Note that these penguins might be different.

(d) All penguins that sing love themselves.

7. Predicate Logic to English

Translate the following sentences to English. Assume the same predicates and domain of discourse as the previous problem.

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

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

(c) $\forall x(\text{Sings}(x) \rightarrow \text{Dances}(x)) \wedge \neg\forall y(\text{Dances}(y) \rightarrow \text{Sings}(y))$

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

8. Tricker Circuits

(a) Draw a truth table for the boolean expression $P(x, y, z)$ that evaluates to false when z is true, and evaluates to $x \oplus y$ when z is false.

(b) Write the DNF (sum of products) form for $P(x, y, z)$.

(c) Draw a circuit to represent $P(x, y, z)$ based on your answer to (b). Your circuit should take x, y, z as input, and only use AND, OR, and NOT gates. Each gate should not take more than two inputs.

9. Tricker Translation

Note: Robbie will go over how to translate more complicated expressions like these in Wednesday's lecture. Feel free to give them a try if you have extra time in workshop.

Express the following sentences in predicate logic. The domain of discourse is movies and actors. You may use the following predicates: Movie(x) ::= x is a movie, Actor(x) ::= x is an actor, Features(x, y) ::= x features y.

(a) Every movie features an actor.

(b) Not every actor has been featured in a movie.

(c) All movies that feature Harry Potter must feature Voldermort.
Hint: You can use "Harry Potter" and "Voldemort" as constants that you can directly plug into a predicate.

(d) There is a movie that features **exactly one** actor.