CSE 390Z: Mathematics for Computation Workshop

Week 3 Workshop Solutions

Conceptual Review

(a) What are CNF and DNF forms?

Solution:

DNF and CNF are two standard forms for producing a Boolean expression, given the Boolean function values. DNF is the "sum of products" form, and CNF is the "product of sums" form.

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

Solution:

A tautology is a statement that is always true. A contradiction is a statement that is always false. A contingency is a statement that is sometimes true, sometimes false.

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

Solution:

Predicate: A function, usually based on one or more variables, that is true or false.

Domain of Discourse: The universe of values that variables come from.

Quantifier: A claim about when the predicate is true. There are two quantifiers. $\forall$ says that the claim is true for all values, and $\exists$ says there exists a value for which the claim is true.

(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"?

Solution:

If we need to restrict something quantified by a "for all", we use **implication**. If we need to restrict something quantifies by an "exists", we use **and**.

For example, suppose the domain of discourse is all animals. We translate "all birds can fly" to $\forall x(\text{Bird}(x) \rightarrow \text{Fly}(x))$. We translate "there is a bird that can fly" to $\exists x(\text{Bird}(x) \wedge \text{Fly}(x))$.

1. Equivalences: Boolean Algebra

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

Solution:

$$
\begin{aligned}
p' + p \cdot q + q' \cdot p &\equiv p' + p \cdot q + p \cdot q' && \text{Commutativity} \\
&\equiv p' + p \cdot (q + q') && \text{Distributivity} \\
&\equiv p' + p \cdot 1 && \text{Complementarity} \\
&\equiv p' + p && \text{Identity} \\
&\equiv p + p' && \text{Commutativity} \\
&\equiv 1 && \text{Complementarity}
\end{aligned}
$$

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

Solution:

$$
\begin{aligned}
(p' + q) \cdot (q + p) &\equiv (p' + q) \cdot q + (p' + q) \cdot p && \text{Distributivity} \\
&\equiv q + (p' + q) \cdot p && \text{Absorption} \\
&\equiv q + qp && \text{Absorption} \\
&\equiv q && \text{Absorption}
\end{aligned}
$$

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 $A(p, q, r)$.

Solution:

DNF: $(p \land \neg q \land r) \lor (\neg p \land q \land r) \lor (\neg p \land q \land \neg r)$

CNF: $(\neg p \lor \neg q \lor \neg r) \land (\neg p \lor \neg q \lor r) \land (\neg p \lor q \lor r) \land (p \lor q \lor \neg r) \land (p \lor q \lor r)$

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

Solution:

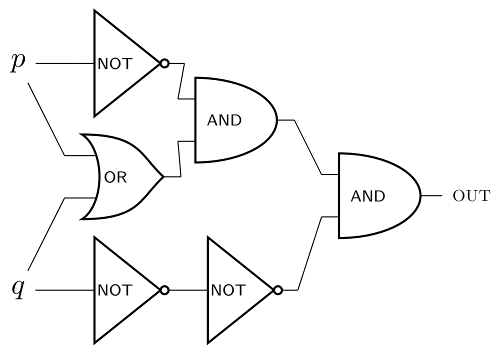
DNF: $(p \land q \land r) \lor (p \land q \land \neg r) \lor (p \land \neg q \land r) \lor (\neg p \land q \land \neg r) \lor (\neg p \land \neg q \land r)$

CNF: $(\neg p \lor q \lor r) \land (p \lor \neg q \lor \neg r) \land (p \lor q \lor r)$

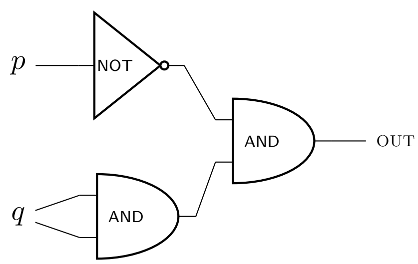
3. Circuits

Convert the following circuits into logical expressions.

(i)



(ii)



Solution:

(i) $((\neg p) \land (p \lor q)) \land \neg\neg q$

(ii) $\neg p \land (q \land q)$

4. Predicate Logic Translation

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

(a) $\forall x($Cat$(x) \land$ Blue$(x))$

Solution:

All animals are blue cats.

(b) $\forall x($Cat$(x) \rightarrow$ Blue$(x))$

Solution:

All cats are blue.

(c) $\exists x($Cat$(x) \land$ Blue$(x))$

Solution:

There exists a blue cat.

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 $\rightarrow$.

Solution:

$\exists x(\neg\text{Cat}(x) \vee \text{Blue}(x))$

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

Solution:

Translation: There exists an animal that is not a cat, or is blue.

The difference: If there was even one non-cat animal in the universe (e.g. a single dog), this condition would be satisfied. Similarly, if there was even one blue animal in the universe, this condition would be satisfied. So, this is a very different condition than "there exists a blue cat".

(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).

Solution:

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

Solution:

True domain: Any set of numbers that includes 0; e.g. all natural numbers.
False domain: Any set of numbers that doesn't include 0; e.g. all integers greater than 0.

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

Solution:

True domain: The set with only the number 0.
False domain: Any set of numbers that includes a value other than 0; e.g. all integers.

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

Solution:

True domain: Any set of numbers that includes two additive inverses; e.g. all integers.
False domain: Any set of numbers that doesn't include any additive inverses; e.g. all positive integers.

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

Solution:

True domain: Any set of numbers that includes 0; e.g. all natural numbers (if $x = 0$, the statement holds for all y).
False domain: Any set of numbers that doesn't include 0; e.g. all integers greater than 0.

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.

Solution:

$\forall x(\text{Dance}(x) \to \neg\text{Sing}(x))$

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

Solution:

$\exists x(\text{Dances}(x) \wedge \text{Sings}(x))$

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

Solution:

$\exists x(\text{Dances}(x)) \wedge \exists x(\text{Sings}(x))$

Note: The variables could be different, and the translation would be equivalent. I.e. this is also correct: $\exists x(\text{Dances}(x)) \wedge \exists y(\text{Sings}(y))$

(d) All penguins that sing love themselves.

Solution:

$\forall x(\text{Sings}(x) \to \text{Loves}(x, x))$

7. Predicate Logic to English

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

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

Solution:

No penguins dance.

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

Solution:

There is a penguin that loves Carol.

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

Solution:

All penguins that sing can dance, but not all penguins that dance can sing.

(d) $\exists x \forall y$(Loves(x, y))

Solution:

There is a penguin that loves all penguins.

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.

Solution:

x	y	z	$P(x, y, z)$
T	T	T	F
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	F

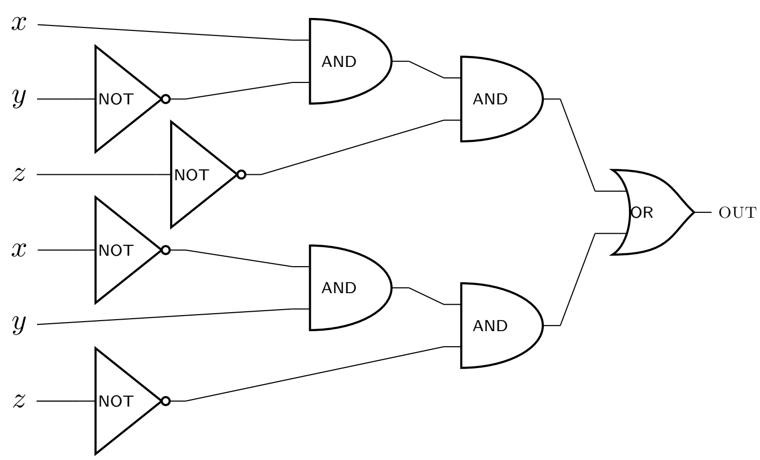
(b) Write the DNF (sum of products) form for $P(x, y, z)$.

Solution:

(x $\wedge\neg$y $\wedge\neg$z) $\vee$($\neg$x $\wedge$y $\wedge\neg$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.

Solution:



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.

Solution:

$\forall x(\text{Movie}(x) \to \exists y(\text{Actor}(y) \land \text{Features}(x, y)))$

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

Solution:

$\neg\forall x(\text{Actor}(x) \to \exists y(\text{Movie}(y) \land \text{Features}(y, x)))$

or, equivalently:

$\exists x(\text{Actor}(x) \land \forall y(\text{Movie}(y) \to \neg\text{Features}(y, x)))$

(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.

Solution:

$\forall x((\text{Movie}(x) \land \text{Features}(x, \text{Harry Potter})) \to \text{Features}(x, \text{Voldemort}))$

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

Solution:

$\exists x\exists y(\text{Movie}(x) \land \text{Actor}(y) \land \text{Features}(x, y) \land \forall z((\text{Actor}(z) \land (z \neq y)) \to \neg\text{Features}(x, z)))$