Recitation 2

Circuits and Sets

Review

In Recitation 1, we started discussing logical equivalences and propositions. Recall some of the terminology for logic, most of which we talked about last week:

1. A **propositional formula** is a function of one or more variables, each of which can be set to true or false, that evaluates to true or false. We call a propositional formula a *proposition* for short.
2. The term **logical expression** is often used synonymously with the word proposition.
3. Two propositions are **logically equivalent** when they have the same truth tables.
4. A proposition is **valid** if it evaluates to true on any choice of inputs; it is true no matter what. That is, a valid proposition is logically equivalent to the expression $(p \vee \neg p)$. This is also called a *tautology.*
5. A proposition is **satisfiable** if it evaluates to true on some choice of inputs. A valid proposition is satisfiable, but so are many propositions which sometimes evaluate to false.
6. If a proposition is **not satisfiable**, it evaluates to false on any choice of inputs; it is false no matter what. That is, it is logically equivalent to the expression $(p \wedge \neg p)$. This is called a *contradiction.*

Normal Forms

We say a proposition is in **DNF (disjunctive normal form)** when it is the disjuction (clauses ORed together) of conjuctions (terms ANDed together).

We say a proposition is in **CNF (conjunctive normal form)** when it is the conjuction (clauses ANDed together) of disjuctions (terms ORed together).

Here's a truth table, and propositions in DNF and CNF which represent it:

p	q	r	?
T	T	T	F
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	T

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

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

If we have an arbitrary truth table, here are two ways we can think about describing it:

- Listing the true rows
- Listing the false rows

Since every row must be either true or false, both of these ways will uniquely describe our truth table.

These two ways correspond to DNF and CNF, respectively. To write a proposition in DNF, we can think about it like this: we find all rows where our proposition should evaluate to true, and we say that we must be in one of those rows. On the other hand, to write a proposition in CNF, we find all rows where our proposition should evaluate to false, and say we are not in any of those rows.

Task: How do we specify that we are in one of the true rows (DNF)? How do we specify that we are not in any of the false rows (CNF)?

Hint : Look at the DNF and CNF representations of the truth tables above. How do they relate to this idea?

Solution:

For DNF, we AND the true variables and negations of the false variables (to be in the row, the inputs must exactly correspond to the row). For CNF, we OR the false variables and the negations of the true variables (to not be in the row, we just need at least one variable to be different)

Task: Write two propositions corresponding to the following truth table: one in DNF and one in CNF.

p	q	r	?
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

Solution:

DNF: $(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$

CNF: $(\neg p \vee q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r)$

Checkpoint 1 — call over a TA!

First-order logic

The language of propositional logic that we've used so far is not expressive enough to talk about real concepts in math or computer science. Consider the proposition, "every Python program runs without crashing." Not very plausible! But, it *implies* the proposition "the Python program `print "Hello world!"` runs without crashing," for instance.

We can't model the connection between these two propositions in propositional logic. They would be represented by two independent propositional letters, p and q.

First-order logic, which includes *quantifiers* and *variables*, lets us go deeper. Let P denote the set of Python programs, and let $R(x)$ mean "program x runs without crashing." Our sentences can then be respectively translated to $\forall x : P, R(x)$ and $R(\texttt{print "Hello world!"})$.

We could even *prove* the implication $\big(\forall x : P, R(x)\big) \to R(\texttt{print "Hello world!"})$.

Your task: translate the sentences below into first-order logic formulas.

Use the following symbols in your translated formulas:

- P: the set of Python programs. (You can use this as the domain for quantifiers.)
- *Predicate symbols* represent propositions, with variable placeholders that can be filled in with any term:
  - $R(x)$: "program x runs without crashing"
  - $T(x)$: "program x terminates" (i.e. doesn't run forever)
  - $C(x, y)$: "program x calls program y"
  - Familiar mathematical relations, like $<$, $\leq$, $=$
- *Function symbols* take in terms as arguments and output new terms:
  - $l(x)$: the number of lines of code in program x
- *Constant symbols* are terms:
  - 0, 1, 2, ... are constant symbols
  - hw is a constant symbol representing the "Hello world" program `print "Hello world!"`.
  - mp is a constant symbol representing the Python program I'm writing right now.

For a first example: we could translate the sentence "There is a program with fewer than 10 lines of code that does not terminate" to $\exists y : P, (l(y) < 10) \wedge \neg T(y)$.

Task

a. Every program with fewer than 10 lines of code terminates.

> **Solution:**
> $\forall y : P, l(y) < 10 \rightarrow T(y)$

b. There is a program that doesn't crash but never terminates.

> **Solution:**
> $\exists y : P, R(y) \wedge \neg T(y)$

c. The "Hello world" program is the shortest program.

> **Solution:**
> $\forall y : P, l(y) \geq l(hw)$ or $\neg \exists y : P, l(y) < l(hw)$

d. The Python program I'm writing right now does not crash, and only calls programs that terminate.

> **Solution:**
> $R(mp) \wedge \forall y : P, C(mp, y) \rightarrow T(y)$

e. Every program calls another program.

> **Solution:**
> $\forall x : P, \exists y : P, C(x, y)$

f. Some program is called by every program. (What's the difference between this and the previous one?)

> **Solution:**
> $\exists y : P, \forall x : P, C(x, y)$

Task

Thought you were done? Try again! It's very common with these types of translations that there are multiple ways to translate any English sentence into logic. Go back and see if you can find an alternate "phrasing" for two of your above answers that seems to capture the same meaning.

Can you see a general pattern in the way you rephrased your answer? Talk it over for a few. (If not, that's okay. We'll come back to this in class soon enough.)

Solution:

A general pattern: $\neg\forall$ is the same as $\exists\neg$. See this, for example, in the answer to c above. This sounds a lot like De Morgan's laws!

Checkpoint 2 — call over a TA!

Set Theory

Defn 1: A **set** is a collection of objects with no repetition or order.

Defn 2: B is a **subset** of A if every element in B is also in A. This is written as $B \subseteq A$.

Defn 3: The **integers** are the set $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$. The **non-negative integers** (also called the natural numbers) are the set $\mathbb{N} = \{0, 1, 2, ...\}$.

Defn 4: A number n is **even** if $n = 2k$ for some $k \in \mathbb{Z}$. A number n is **odd** if $n = 2k + 1$ for some $k \in \mathbb{Z}$.

Defn 5: A number n is **rational** if $n = \frac{a}{b}$ for some $a, b \in \mathbb{Z}$, where $b \neq 0$.

Defn 6: $\mathcal{P}(A)$, called the **power set** of A, is the set of all subsets of A.

Defn 7: $A \cup B$ denotes the **union** of sets A and B. This contains all the elements from A, and all of the elements from B.

Defn 8: $A \cap B$ denotes the **intersection** of sets A and B. This contains only the elements that appear in both of the sets.

Defn 9: $A \setminus B$ denotes the **difference** of sets A and B. This contains elements that appear in A but not B.

Defn 10: $\overline{A}$ denotes the **complement** of A relative to some universal set U. $\bar{A} = U - A$, that is, it is everything except what is in A.

Defn 11: $|A|$ denotes the **cardinality** of A, which is a count of the number of elements contained in A.

Defn 12: The symbol $\forall$ means **for all**.

Defn 13: The symbol $\exists$ means **there exists**.

Answer the following questions. Discuss your answers!

Task

a. True or False: A is an arbitrary set. Answer true only if the statement is always true. That is, answer true only if for any possible set A, the statement is true.

   i. $A \subseteq A$

   Solution:
   (T)

   ii. $\{\} \subseteq A$

   Solution:
   (T)

   iii. $\{\} \in A$

   Solution:
   (F)

b. True or False: $\mathbb{N} \subseteq \mathbb{Z}$

Solution:
(T)

c. $\{0, 1, 9\} \subseteq \mathbb{N}$

Solution:
(T)

d. True or False: $\{-1.5, 9\} \subseteq \mathbb{Z}$

Solution:
(F)

e. Let $\mathbb{Q}$ be the set of rational numbers.

   i. $\mathbb{Q} \cap \mathbb{N} = \mathbb{N}$

   Solution:
   (T)

ii. $\mathbb{Q} \cup \mathbb{N} = \mathbb{R}$

> **Solution:**
> (F)

f. True or False: S is the set of blue flowers in Wednesday Addams's bouquet. G is the set of all colors of flowers in her bouquet. Wolfsbane is a blue flower in her bouquet.

i. $S \subseteq G$

> **Solution:**
> (T)

ii. Wolfsbane $\subseteq S$

> **Solution:**
> (F)

iii. Wolfsbane $\in S$

> **Solution:**
> (T)

iv. $\{\text{Wolfsbane}\} \subseteq G$

> **Solution:**
> (T)

g. True or False: If $A = \{1, 2, 4\}$ then $\{2, 4\} \in \mathcal{P}(A)$

> **Solution:**
> (T)

h. True or False: A is a set, and $\mathcal{P}(A)$ is the set of all subsets of A. Answer true only if the statement is always true.

i. $A \in \mathcal{P}(A)$

> **Solution:**
> (T)

ii. $A \subseteq \mathcal{P}(A)$

Solution:
(F)

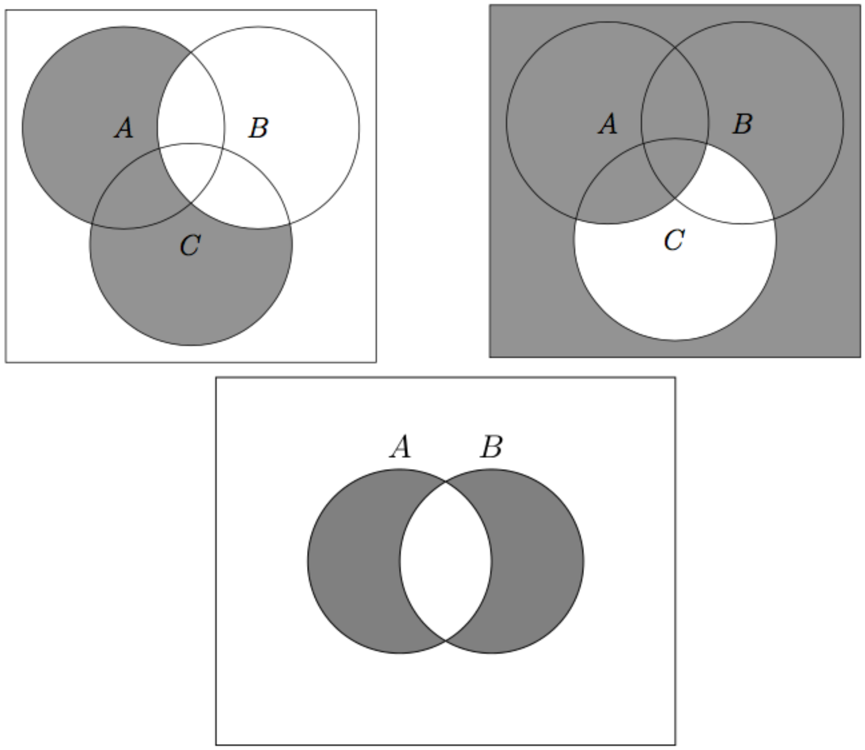
iii. $\emptyset \in \mathcal{P}(A)$

Solution:
(T)

iv. $\emptyset \subseteq \mathcal{P}(A)$

Solution:
(T)

i. In each of the following Venn diagrams, A, B, and C are sets and are assumed to be subsets of a universal set (denoted by the rectangle). Write a set algebraic expression (i.e. one involving union, intersection, difference, and complement) in terms of A, B, and C for each shaded in region.



Solution:

$(A \cup C) - B$
$\overline{C} \cup A$
$(A \cup B) - (A \cap B)$

j. *Optional:* Call a the cardinality of A and b the cardinality of B. Call s the cardinality of $A \cap B$. For the third picture, what is the cardinality of the set formed from the expression you derived?

Solution:

Solution: $a + b - 2s$

Finall checkoff — call over a TA!