Name:

CSE 1400 Applied Discrete Mathematics

Spring 2015 Week 3 Key

1. Let's reason about Boolean logic. Consider one of De Morgan's laws:

$$\neg(\text{P} \wedge \text{Q}) \equiv \neg\text{P} \vee \neg\text{Q}$$

If the left-hand side is `True`, then $\text{P} \wedge \text{Q}$ is `False`. That means at least one of P or Q is `False`. And that means at least one of $\neg\text{P}$ or $\neg\text{Q}$ is `True`, and therefore, $\neg\text{P} \vee \neg\text{Q}$ is `True`.

In the other case, if the left-hand side is `False`, then $\text{P} \wedge \text{Q}$ is `True`. That means both P and Q are `True`. And that means both of $\neg\text{P}$ and $\neg\text{Q}$ are `False`, and therefore, $\neg\text{P} \vee \neg\text{Q}$ is `False`.

Mimic what I just wrote to argue that the second of De Morgan's laws is `True`.

$$\neg(\text{P} \vee \text{Q}) \equiv \neg\text{P} \wedge \neg\text{Q}$$

Answer: If the left-hand side is `True`, then $\text{P} \vee \text{Q}$ is `False`. That means both P and Q are `False`. And that means both $\neg\text{P}$ or $\neg\text{Q}$ are `True`. Therefore, $\neg\text{P} \wedge \neg\text{Q}$ is `True`.

On the flip side, if the left-hand side is `False`, then $\text{P} \vee \text{Q}$ is `True`. That means at least one of P or Q is `True`. And that means at least one of $\neg\text{P}$ and $\neg\text{Q}$ is `False`. Therefore, $\neg\text{P} \wedge \neg\text{Q}$ is `False`.

Congratulations! You just gave a proof!

2. Here's the mechanical, truth table, proof of De Morgan's law. Rename the left and right-hand sides of De Morgan's laws $\text{R} = \neg(\text{P} \wedge \text{Q})$ and $\text{S} = \neg\text{P} \vee \neg\text{Q}$. My argument had two steps:

   (a) $\text{R} \Rightarrow \text{S}$. If R was `True`, then S was also `True`.

   (b) $\neg\text{R} \Rightarrow \neg\text{S}$. If R was `False`, then S was also `False`.

Fill in the truth table for $(\text{R} \Rightarrow \text{S}) \wedge (\neg\text{R} \Rightarrow \neg\text{S})$ to conclude that it be reduced to $\text{R} \equiv \text{S}$.

Answer: Here is the truth table. Notice the result, that AND($\wedge$) column is identical to EQUIVALENT($\equiv$).

R	S	$(\text{R} \Rightarrow \text{S})$	$\wedge$	$(\neg\text{R} \Rightarrow \neg\text{S})$
0	0	1	1	1
0	1	1	0	0
1	0	0	0	1
1	1	1	1	1

Note that $\neg\text{R} \Rightarrow \neg\text{S}$ is equivalent to $\text{S} \Rightarrow \text{R}$. This *equivalence* is called contraposition.

3. Last week you constructed truth tables to demonstrate that

$$((\text{P} \wedge \text{Q}) \Rightarrow \text{R}) \equiv (\text{P} \Rightarrow (\text{Q} \Rightarrow \text{R}))$$

This equivalence is called currying, named for Haskell Curry who explained its power in logic and computation: It reduces the computation of a function of many arguments to the composition of many functions of a single argument.

Write an explanation of the equivalence in English. To do this, consider, in sequence, the possible values of P, Q, and R.

Answer: Suppose P is `False`. Then the left-hand and right-hand sides of the equivalence are both `True`. Both sides have the form

$$\texttt{False} \Rightarrow \text{whatever} \quad \text{which evaluates to } \texttt{True}$$

It does not matter what you derive from a `False` premise. The implication is `True`.

Suppose P is `True`. Then you must examine the value of Q.

If Q is `False` then the left-hand and right-hand sides of the equivalence have the same value. The left-hand side has the form

$$\texttt{False} \Rightarrow \text{whatever} \quad \text{which evaluates to } \texttt{True}$$

and the right-hand side has the form

$$\texttt{True} \Rightarrow (\texttt{False} \Rightarrow \text{whatever}) \equiv \texttt{True} \Rightarrow \texttt{True} \quad \text{which is also } \texttt{True}$$

Now, what if P and Q are `True`, then the truth of the the left-hand and right-hand sides of the equivalence depends on R.

The left-hand side has the form

$$\texttt{True} \Rightarrow \text{R} \quad \text{which evaluates to R}$$

and the right-hand side has the form

$$\texttt{True} \Rightarrow (\texttt{True} \Rightarrow \text{R}) \quad \text{which also reduces to R}$$

Therefore, in all possible cases the exprssions on the left and right in the equivalence $\texttt{True} \Rightarrow (\texttt{False} \Rightarrow \text{whatever}) \equiv \texttt{True} \Rightarrow \texttt{True}$ which is also `True` have identical values.

4. Resolution is another powerful inference tool. It says

$$((\text{P} \Rightarrow \text{Q}) \wedge (\neg\text{P} \Rightarrow \text{R})) \Rightarrow (\text{Q} \vee \text{R})$$

   (a) Construct a truth table to demonstrate resolution is a valid conditional.
   Answer:

P	Q	R	(P ⇒ Q)	∧	(¬P ⇒ R)	⇒	Q ∨ R
0	0	0	1	0	0	1	0
0	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0
1	0	1	0	0	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

   (b) Give an English interpretation of resolution.
   Answer: Resolution states this: If you know both $\text{P} \Rightarrow \text{Q}$ and $\neg\text{P} \Rightarrow \text{R}$ are `True`, then you know one of Q or R must be `True`. That's because one of P or ¬P must be `True`, and so, at least one of Q or R must be `True`.

   (c) Look at your truth table. See that there are only two rows where $(\text{P} \Rightarrow \text{Q}) \wedge (\neg\text{P} \Rightarrow \text{R})$ and $\text{Q} \vee \text{R}$ fail to be equivalent. Explain these two cases.
   Answer: When P is `False`, Q is `True`, and R is `False`, $\neg\text{P} \Rightarrow \text{R}$ is not `True`. Similarly, when P is `True`, Q is `False`, and R is `True`, $\text{P} \Rightarrow \text{Q}$ is not `True`.

5. Set theory and Boolean logic share a common structure: They, with their operations, are both Boolean algebras. One commonality is the count of basic things.

(a) Review your answers to question 7 from last week's homework:

How many bit strings of length n are there?

Answer: There are 2^n bit strings.

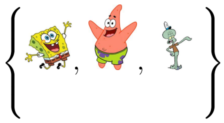
How many truth assignments can be made on n Boolean variables?

Answer: There are are 2^n truth assignments.

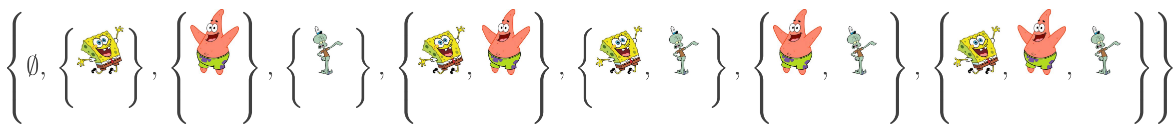
A complete truth table for an n variables Boolean expression will have how many rows?

Answer: There will be 2^n rows. Again, the same expression.

(b) How many subsets does the set

$\mathbb{SBSP} =$  have?

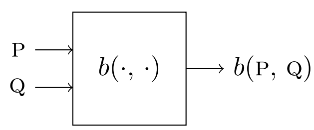
Answer: The set of all subsets of $\mathbb{SBSP}$, called the power set of $\mathbb{SBSP}$ is

$2^{|\mathbb{SBSP}|} =$ 

(c) How many subsets does an n element set have?

Answer: There are 2^n subsets of an n element set. Subsets of an n-element set can be represented as a bit-string of length: The value of a bit in the string is set to 1 or 0 depending on whether or not the corresponding element is in or not in the subset.

6. In logic you thought about passing truth assignments through a function. For instance, you learned about some 2-input Boolean functions like AND, OR, EQUIVALENT, EXCLUSIVE-OR, and IMPLIES. These functions can be visualized using a diagram such as this.



Where $b(\cdot,\, \cdot)$ is a generic name for a 2-input Boolean function.

How many truth assignments does a two-input Boolean function have?

Answer: There are four combinations of input:

$$(\text{P},\ \text{Q}) \in \{(0,\ 0),\ (0,\ 1),\ (1,\ 0),\ (1,\ 1)\}$$

How many two-input Boolean functions are there?

Answer: A function is defined by mapping input to output. There are $4 = 2^2$ input combinations for a two-input function. And, there 2 output choices for each combination. Therefore, there are $2^4 = 2^{2^2}$ two-input Boolean functions.

How many n-input Boolean functions are there?

Answer: There are 2^{2^n} n-input Boolean functions. A function is defined by mapping input to output. There are 2^n input combinations for an n-input function. And, there are 2 output choices for each combination. Therefore, there are 2^{2^n} n–input Boolean functions.

Truth tables are another way to describe Boolean functions. A complete truth table for an n variables Boolean expression has how many rows?

How many bit strings of length 2^n are there?

Answer: There are 2^{2^n} bit strings of length 2^n.

In how many ways can 2^n bit strings be map to `True` or `False`?

Answer: There are 2^{2^n} ways to map 2^n expressions to `True` or `False`.

In a truth table with 2^n rows, how many columns of 0's and 1's can be constructed?

Answer: There are 2^{2^n} ways to fill a column in a truth table with 2^n rows.

7. The conditional operator $\texttt{P} \Rightarrow \texttt{Q}$ is equivalent to $\neg\texttt{P} \lor \texttt{Q}$, which can be converted mechanically into set notation $\neg\mathbb{P} \cup \mathbb{Q}$ (don't think of $\mathbb{P}$ and $\mathbb{Q}$ as the sets of prime and rational numbers: Here they just name matching the logic notation)

   Fill in the truth table Answer:

$x \in \mathbb{P}$	$x \in \mathbb{Q}$	$x \in \neg\mathbb{P}$	$\lor$	$x \in \mathbb{Q}$
0	0	1	1	0
0	1	1	1	1
1	0	0	0	0
1	1	0	1	1

   Make a conjecture about how to interpret the Boolean operator $\Rightarrow$ in the context of sets.

   Answer: The conditional operator acts like subset $\subseteq$. The statement $\mathbb{P} \subseteq \mathbb{Q}$ has this truth table.

$x \in \mathbb{P}$	$x \in \mathbb{Q}$	$x \in \mathbb{P} \Rightarrow x \in \mathbb{Q}$
0	0	1
0	1	1
1	0	0
1	1	1

   Subset $\mathbb{X} \subseteq \mathbb{Y}$ says: If $a \in \mathbb{X}$, then $a \in \mathbb{Y}$. On the other hand, if $a \notin \mathbb{X}$, then we don't necessarily know if a is in $\mathbb{Y}$ or not. That is, implication in Boolean logic acts like subset in set theory.

8. Set theory and Boolean logic share a common structure: But, notationally set theory uses round symbols while pointy symbols are used in logic.

Match the natural equivalences between set theory and Boolean algebra notation. What are the common names for each symbol? Be certain you understand the meanings of the names and their context.

Logic symbols

(a) $\wedge$

(b) {True, False}

(c) $\vee$

(d) $\Rightarrow$

Set symbols

(a) $\in$

(b) $\cup$

(c) $\cap$

(d) $\subseteq$

Answer: AND$\wedge$ acts like INTERSECT$\cap$. $\wedge$ :: BOOL $\times$ BOOL $\mapsto$ BOOL, while $\cap$ :: SET $\times$ SET $\mapsto$ SET.

OR$\vee$ acts like UNION$\cap$. OR $\Rightarrow$ acts like SUBSET$\subseteq$.

The ELEMENTor MEMBERfunction has a different type from the rest. It maps a value and a set to True or False. $\in$:: VALUE $\times$ SET $\mapsto$ BOOL. That is, $\in$ is most like {True, False}. It defines a *relation* that is or is not True.

9. Let $\mathbb{EVEN} = \{0, 2, 4, 6, 8\}$, $\mathbb{ODD} = \{1, 3, 5, 7, 9\}$ and $\mathbb{PRIME} = \{2, 3, 5, 7\}$ be the *even*, *odd* and *prime* digits. Compute the following set operations over the universe of digits $\mathbb{U} = \mathbb{D} = \{0, 1, 2, \ldots, 8, 9\}$.

(a) $\neg\mathbb{EVEN}$
Answer: $\neg\mathbb{EVEN} = \mathbb{ODD}$

(b) $\neg\mathbb{PRIME}$
Answer: $\neg\mathbb{PRIME} = \{0, 1, 4, 6, 8, 9\}$

(c) $\mathbb{ODD} \cup \mathbb{PRIME}$
Answer: $\mathbb{ODD} \cup \mathbb{PRIME} = \{1, 2, 3, 5, 7, 9\}$

(d) $\mathbb{EVEN} \cup \mathbb{PRIME}$
Answer: $\mathbb{EVEN} \cup \mathbb{PRIME} = \{0, 2, 3, 4, 5, 6, 7, 8\}$

(e) $\mathbb{ODD} \cap \mathbb{PRIME}$
Answer: $\mathbb{ODD} \cap \mathbb{PRIME} = \{3, 5, 7\}$

(f) $\mathbb{EVEN} \cap \mathbb{PRIME}$
Answer: $\mathbb{EVEN} \cap \mathbb{PRIME} = \{2\}$

(g) $\mathbb{EVEN} \cap \mathbb{ODD}$
Answer: $\mathbb{EVEN} \cap \mathbb{ODD} = \emptyset$

(h) $\mathbb{ODD} \cap \neg\mathbb{PRIME}$
Answer: $\mathbb{ODD} \cap \neg\mathbb{PRIME} = \{1, 9\}$

10. Reasoning about nothing can be difficult. Reasoning about everything is difficult too. Which of the following statements about the empty set are True and which are False? Explain your answers. Recall $\in$ means "is an element of" and $\subseteq$ means "is a subset of."

(a) $\emptyset = \{\emptyset\}$
Answer: This is False. The empty set $\emptyset$ contains no elements. The set $\{\emptyset\}$ has one element, which happens to be $\emptyset$.

(b) $\emptyset \in \{\emptyset\}$
Answer: This is True. The empty set is an element in the set containing the empty set.

(c) $\emptyset \in \emptyset$
Answer: This is False. The empty set is an element in the empty set. The empty set has no elements.

(d) $\{\emptyset\} \in \emptyset$
Answer: This is False. The set containing the empty set is not an element in the empty set. The empty set has no elements at all.

(e) $\{\emptyset\} \in \{\emptyset\}$
Answer: This is False. The set containing the empty set is not an element in the set containing the empty set. But $\{\emptyset\} \in \{\{\emptyset\}\}$ is True.

(f) $\emptyset \subseteq \{\emptyset\}$
Answer: This is True. The conditional $x \in \emptyset \Rightarrow x \in \emptyset$ is logically True, and it defines the meaning of $\emptyset \subseteq \emptyset$. The empty set is a subset of every set, including itself.

(g) $\emptyset \subseteq \emptyset$
Answer: This is True. The conditional $x \in \emptyset \Rightarrow x \in \emptyset$ is logically True, and it defines the meaning of $\emptyset \subseteq \emptyset$. The empty set is a subset of every set, including itself.

(h) $\{\emptyset\} \subseteq \emptyset$
Answer: This is False. The set $\{\emptyset\}$ is not empty. Therefore it is not a subset of $\emptyset$.

(i) $\{\emptyset\} \subseteq \{\emptyset\}$

Answer: This is `True`. A subset is always a subset of itself.

Total Points: 0

2015-01-26 to 2015-01-30