L04LogicalEquivalences2

January 31, 2017

1 Lecture 4

More examples please: on the board More questions: 4
logical equivalences: converse, contrapositive, inverse etc. - show equivalences
other equivalences list examples use, fasteer than truthtable example de morgan's
Analogies with set theory propositional functions sets, truth sets
Reading on sets: Rosen 2.1 and 2.2
Homework, HTPI from last year + functionally complete + more?

In [11]:
```
%matplotlib inline
%config InlineBackend.figure_format='retina'
# import libraries
import numpy as np
import matplotlib as mp
import pandas as pd
import matplotlib.pyplot as plt
import laUtilities as ut
import slideUtilities as sl
import demoUtilities as dm
import prettytable
from IPython.display import Image
from IPython.display import display_html
from IPython.display import display
from IPython.display import Math
from IPython.display import Latex
from truths import Truths
#reload(dm)
#reload(ut)
#print('')
```

In [12]:
```
%%html
<style>
 .container.slides .celltoolbar, .container.slides .hide-in-slideshow {
    display: None ! important;
}
</style>
```

<IPython.core.display.HTML object>

```
In [ ]:
```

1.1 Logical Equivalences

- Useful laws to simplify and evaluate propositions:
  - Examples for Disjunction and Conjunctions:
    * Associative Laws
    * Distribute Laws
    * Commutative Laws

Tip 1: You do not need to remember the names of the other laws. **Tip 2:** We write $P \equiv Q$ to say that P and Q are equivalent propositions. Note that this is slightly different from $P \iff Q$.

The statement $P \equiv Q$ tells us that $P \iff Q$ is a tautology, e.g., P and Q always take the same truth values. However: - $P \iff Q$ is a valid proposition in our propositional logic - $P \equiv Q$ is not a proposition. It is something we do outside of the logic system.

1.1.1 Today:

1. More equivalences:
   - involving $\vee, \wedge$ and $\neg$: De Morgan's Laws
   - involving conditional statements
2. Predicates, truth sets:
   - set operations
   - analogy between set theory and propositional logic

1.2 Practice Question

Question: Which of the following propositions is logically equivalent to

$$\neg(P \wedge (Q \vee \neg P)?$$

a. \$ P $\vee Q$; \$
   $\neg P \vee Q$;

bc. $P \vee \neg Q$;

d. $\neg P \vee \neg Q$;

Answer: d.
Truth table. How to do it with Laws?

1.3 De Morgan's Laws

Example:

`Both the Yankees and the Red Sox lost last night.`

Its negation is:

`Either the Yankees or the Red Sox won last night.`

1.4 De Morgan's Laws

- 1st Law: Negation of Conjunction is Disjunction of Negations

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q.$$

- 2nd Law: Negation of Disjunction is Conjunction of Negations

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q.$$

Tip: You don't need to remember which one is 1st or 2nd.

1.5 Verifying De Morgan's Laws by Truth Table

Idea: The truth table of the $\wedge$ operator is the negation of the truth table of the $\vee$ operator.

1.6 Equivalences involving conditional statements

This is a *conditional* statement: $P \implies Q$
this is its *converse*: $Q \implies P$
this is its *inverse*: $\neg P \implies \neg Q$
this is its *contrapositive*: $\neg Q \implies \neg P$

1.7 Practice Question

Which one of the following statements is logically equivalent to
> If at least 10 people are there, then the lecture will be given. ?

a. If there are fewer than 10 people, then the lecture will not be given.

b. If the lecture is not given, it means that there were fewer than 10 people there.

c. If the lecture is given, at least 10 people were there.

d. The lecture will be given if and only if at least 10 people are there.

b. Contrapositive

1.8 Equivalence of contrapositive via laws

In this case of chain of equivalences is more elegant than a truth table.

- How about converse and inverse?

  Show scheme.

1.9 Practice Question

For which of the following statements is the converse true? [Multiple Answers!]

a. If $x > 3$ and $y < 2$, then $x > y$.

b. If $x > y$ then $x - y > 0$.

c. If $x^2 = y^2$, then $x = y$.

d. If $x > y$, then $x = y$.

e. If $x \geq y$, then $x = y$.

1.9.1 We are done with logical equivalences.

Going forward, remember the different ways to show logical equivalence:

a. truth table

b. by applying laws (i.e., known logical equivalence)

c. combining both

1.10 Beyond propositional variables

The propositional logic we covered so far is not very interesting: - we are never looking inside the propositional variables, - does not convey a mathematical meaning.

Next, we'll introduce some mathematical structure. For this purpose, I am assuming you are familiar with basic set notation: - $x \in A$ means x is element of set A, i.e., x belongs to A. - $A \subseteq B$ means that A is a subset of B. - $A = \{x \in \mathbb{R} | x > 0\}$ defines A as the set of positive real numbers.

1.11 Predicates

To use propositiona logic to talk about mathematical structures, we start by introducing **predicates**:

Definition: Predicates are propositions that contain variables.

Examples: - P(x): The integer x is prime. - Q(z): \$ z $\geq 0.$\$

The truth value of these propositions depends on the setting of the variables: - P(10) is false, but P(11) is true. - Q(1) is true, but Q(-1) is false.

Idea: As the notation suggests, you should think of $P(x)$ as a function of x:

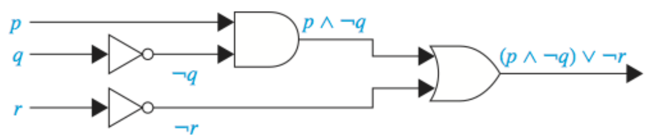
P takes as input an element x from a *universe* U and returns a proposition $P(x)$.

Note: Be careful that you understand which universe $P(x)$ operates over.
Examples: - P(x): The integer x is prime. Universe $U = \mathbb{Z}$

- Q(z): \$ z $\geq 0.\$ Universe U =$

TRUTH SETS SETS OPERATIONS QUANTIFIERS

```
In [13]: # image credit: Kenneth Rosen "Discrete Mathematics and Its Applicati
         sl.hide_code_in_slideshow()
         display(Image("images/L02/circuit.png", width=350))
```




```
In [14]: a = Truths(['P', 'Q'], [,'P or Q', 'R or Q']).__str__();
         a = a.replace('+','|');
         a = a.replace('1','T');
         a = a.replace('0','F');
         print(a);
```

```
  File "<ipython-input-14-a7f75f3a068c>", line 1
    a = Truths(['P', 'Q'], [,'P or Q', 'R or Q']).__str__();
                            ^

SyntaxError: invalid syntax
```

1.12 Mathematical Proofs

In mathematical proofs (and logical deduction in general), we start with some assumptions (i.e., premises) and derive some theses (i.e., conclusions).
A MATHEMATICAL PROOF:
ASSUMPTION $P \implies P_1 \implies P_2 \implies P_3 \implies P_4 \implies C$ **THESIS**
To complete the proof, we must verify that all the implications are valid (i.e., true). That means, we must check that indeed: - $P \implies P_1$, - $P_1 \implies P_2$, - $P_2 \implies P_3$, - $P_3 \implies P_4$, ... Each of these steps may require a mathematical proof itself.
Where does this stop?

  - Axioms

1.13 Lecture 2 Add-on: Conditional relations between two propositions

Given two propositions P and Q, we have seen that they can be conditionally related in different ways.

```
In [ ]:
```