CMPSCI 250: Introduction to Computation

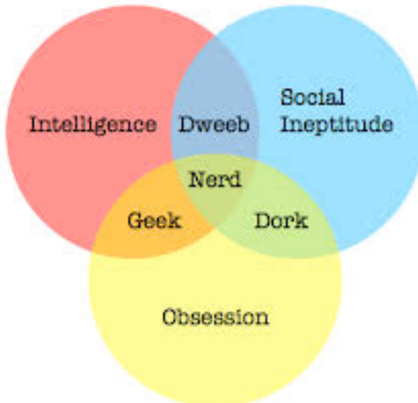
Lecture #3: Set Operations and Truth Table Proofs
David Mix Barrington
9 September 2013

Set Operations and Truth Tables

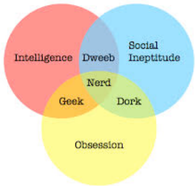
- Venn and Carroll Diagrams
- Set Operations
- Propositions About Sets
- The Setting for Propositional Proofs
- How to Do a Truth Table Proof
- A Truth Table Proof Example

Venn Diagrams

- Here's a way to describe a group of sets.



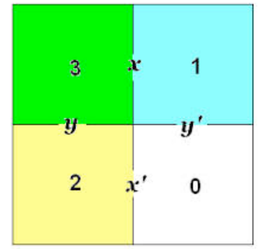
Venn Diagrams



- The three large sets each divide the type into two groups: the elements in it and those not in it.
- This creates $2^3 = 8$ total groups, from the three choices.
- This **Venn Diagram** has seven colored regions, and an eighth white region in none of the sets.

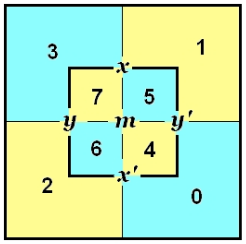
Carroll Diagrams

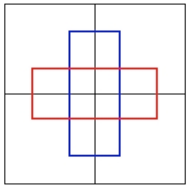
- Lewis Carroll (author of Alice in Wonderland) had his own diagrams he liked better than Venn's.
- This diagram represents the four combinations of being in set x or not, and being in set y or not. For example, region 2 is in y but not in x.
- Unlike Venn, he treats the four regions equally.



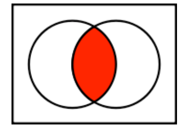
More Carroll Diagrams

- In the top diagram we represent three sets, with m the set inside the central box. Region 5 is in m and x but not in y.
- Binary for 5 is 101, with the three bits for yes-m, no-y, yes-x.
- The bottom diagram represents the 16 regions for four sets.



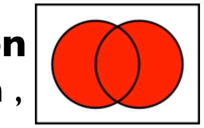


Set Operations



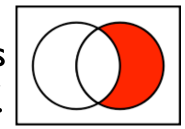
$X \cap Y$

- We have a number of **binary operations** on sets, that take two sets as input and give one set as output.
- If X and Y are sets, their **intersection** $X \cap Y$ is the set of all elements in both , and their **union** $X \cup Y$ is the set of all elements in either X or Y.



$X \cup Y$

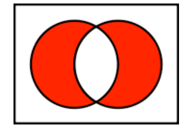
- The **relative complement** $X \setminus Y$ is the set of all elements in X but not in Y.



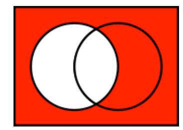
$Y \setminus X$

Two More Set Operations

- The **symmetric difference** X ΔY is the set of elements that are in either X or Y, but not both.
- The **complement** of X, written as X with a line over it, is the set of all elements in the **universe** (or data type) that are not in X.

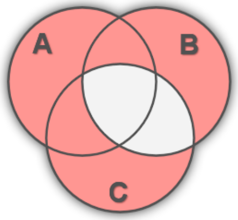


X ΔY



X complement

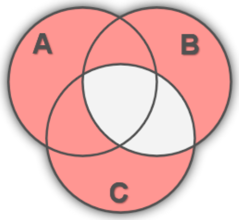
Practice Clicker Question #1



- What set is denoted in red?
- (a) (A ∪ B ∪ C) ∖ (B ∩ C)
- (b) (A ∩ B ∩ C) ∖ (B ∩ C)
- (c) (A ∪ B ∪C) ∖ (B Δ C)
- (d) A ∪ (B Δ C)

Answer #1

- What set is denoted in red?
- (a) *(A ∪ B ∪ C) ∖ (B ∩ C)*
- (b) (A ∩ B ∩ C) ∖ (B ∩ C)
- (c) (A ∪ B ∪C) ∖ (B Δ C)
- (d) A ∪ (B Δ C)



Propositions About Sets

- Given two sets X and Y, we can form the propositions $X = Y$ and $X \subseteq Y$. We can also use the = and $\subseteq$ operators on more complicated sets formed with the set operators, for example $(X \setminus Y) \cap (Y \setminus X) = \emptyset$.
- This last statement is an example of a set identity because it is true no matter what the sets X and Y are. Since every element of $X \setminus Y$ is in X, and none of the elements of $Y \setminus X$ are in X, no element could be in both.

Membership Statements

- Equality and subset statements about sets are actually compound propositions involving **membership statements** for the original sets.
- For example, $X = Y$ means that for any object z of the correct type, the propositions $z \in X$ and $z \in Y$ are either both true or both false, so that "$z \in X \leftrightarrow z \in Y$" is true.
- Similarly, $X \subseteq Y$ means that for any z, $z \in X$ implies $z \in Y$, so we have "$z \in X \rightarrow z \in Y$".

Set Identities With Set Operators

- A set statement like $(X \setminus Y) \cap (Y \setminus X) = \varnothing$, using set operations and the equality or subset operator, can be translated into a compound proposition.
- We want to say $z = (X \setminus Y) \cap (Y \setminus X) \leftrightarrow z \in \varnothing$.

  But the statement on the left of the $\leftrightarrow$ can be

  simplified, to $z \in (X \setminus Y) \wedge z \in (Y \setminus X)$.
- Using the definition of $\setminus$, this can be further simplified to $(z \in X \wedge \neg (z \in Y))$ ^ $(z \in Y$ ^ $\neg(z \in X))$.

Using Variables for Each Set

- If we define the boolean x to mean $z \in X$ and the boolean y to mean $z \in Y$, we can rewrite the whole statement "$(z \in X \wedge \neg (z \in Y))$ ^ $(z \in Y$ ^ $\neg(z \in X))$" as $(x \wedge \neg y) \wedge (y \wedge \neg x) \leftrightarrow 0$,

  where we use 0 to mean "false".
- This compound proposition is a tautology.
- In the same way we can translate any set statement, because each set operation corresponds exactly to a boolean operation on membership statements.

Practice Clicker Question #2

- Let r denote the proposition "$x \in R$", s denote "$x \in S$", and t denote "$x \in T$". Which of the following is denoted by $r \wedge (s \oplus t)$?
- (a) $x \in (R \cap S) \Delta T$
- (b) $x \in R \cup (S \Delta T)$
- (c) $x \in R \cap (S \Delta T)$
- (d) $x \in (R \cup S) \Delta T$

Answer #2

- Let r denote the proposition "$x \in R$", s denote "$x \in S$", and t denote "$x \in T$". Which of the following is denoted by $r \wedge (s \oplus t)$?
- (a) $x \in (R \cap S) \Delta T$
- (b) $x \in R \cup (S \Delta T)$
- (c) *$x \in R \cap (S \Delta T)$*
- (d) $x \in (R \cup S) \Delta T$

The Setting for PropCalc Proofs

- The propositional calculus lets us form compound propositions from atomic propositions, and then ask questions about them.
- Is a given statement P a **tautology**? If we know that a **premise** statement P is true, does that guarantee that another **conclusion** statement C is also true? Given two statements P and Q, are they **equivalent**?
- Verifying tautologies solves all three of these questions, because they ask whether P, $P \rightarrow C$, and $P \leftrightarrow Q$ respectively are tautologies.

The Bigger Picture

- In this lecture we'll see how to verify a tautology with a **truth table**.
- Next week we'll see how to verify that an implication or an equivalence is a tautology with a **deductive sequence proof** or an **equational sequence proof**.
- Sequence proofs can be much shorter than the corresponding truth tables, but they require creativity to produce.

How to Do a Truth Table Proof

- The idea of a truth table proof is that if we have k atomic propositions, there are 2^k possible settings of the truth values of those propositions. If a given compound proposition is true in all of those cases, it is a tautology.
- We need to evaluate the compound proposition systematically, in all the cases. We begin by listing the cases, which we can do by counting in binary from 0 to $2^k - 1$, which is from 00...0 to 11...1. (This is much less error-prone than trying to get all the cases in some arbitrary order.)

How to Do a Truth Table Proof

- The basic idea is that under each symbol of the compound proposition, we will have a column of 2k 0's and 1's to represent the values, in each case, of the compound proposition associated with that symbol.
- We begin with the occurrences of the variables, then calculate new columns in the order that operations are used to evaluate the compound proposition.

A Truth Table Example

- Let's take the formula (x ∧ ¬ y) ∧ (y ∧ ¬ x) ↔ 0. There are four cases 00, 01, 10, and 11, where the first bit is the truth value of x and the second that of y. We write the correct column under each occurrence of a variable. We also write a column of all 0's under the 0, since this symbol always has the value 0.

```
x y | (x ∧ ¬ y) ∧ (y ∧ ¬ x) ↔  0
-------------------------------
0 0    0      0      0      0      0
0 1    0      1      1      0      0
1 0    1      0      0      1      0
1 1    1      1      1      1      0
```

Continuing The Example

- Next we fill in the columns for the ¬ operations:

```
x y | (x ∧ ¬ y) ∧ (y ∧ ¬ x) ↔  0
--------------------------------
0 0    0   1 0     0   1 0     0
0 1    0   0 1     1   1 0     0
1 0    1   1 0     0   0 1     0
1 1    1   0 1     1   0 1     0
```

Continuing The Example

- Then the two ∧ operations inside the parentheses:

```
x y | (x ∧ ¬ y) ∧ (y ∧ ¬ x) ↔  0
--------------------------------
0 0    0 0 1 0     0 0 1 0     0
0 1    0 0 0 1     1 1 1 0     0
1 0    1 1 1 0     0 0 0 1     0
1 1    1 0 0 1     1 0 0 1     0
```

Continuing The Example

- Then the last ∧ operation:

```
x y | (x ∧ ¬ y) ∧ (y ∧ ¬ x) ↔  0
--------------------------------
0 0    0 0 1 0  0  0 0 1 0     0
0 1    0 0 0 1  0  1 1 1 0     0
1 0    1 1 1 0  0  0 0 0 1     0
1 1    1 0 0 1  0  1 0 0 1     0
```

Finishing the Example

- And finally the ↔ operation. Since this final

  column is all 1's, we have shown that the original compound proposition is a tautology.

```
x y | (x ∧ ¬ y) ∧ (y ∧ ¬ x) ↔  0
--------------------------------
0 0    0 0 1 0  0  0 0 1 0  1  0
0 1    0 0 0 1  0  1 1 1 0  1  0
1 0    1 1 1 0  0  0 0 0 1  1  0
1 1    1 0 0 1  0  1 0 0 1  1  0
```

One More Venn Diagram