

Week 7 - Friday

COMP 2230

Last time

- Set definitions
- Set properties

Questions?

Assignment 4

Logical warmup

- You have 15 bags
- How many marbles do you need so that you can have a different number of marbles in each bag?

Disproofs, Algebraic Proofs, and Russell's Paradox

Three-sentence summary

Disproofs and Algebraic Proofs

Disproving a set property

- Like any disproof for a universal statement, you must find a counterexample to disprove a set property
- For set properties, the counterexample must give a specific example of sets for each set in the claim

Counterexample example

- **Claim:** For all sets A , B , and C , $(A - B) \cup (B - C) = A - C$
- Find a counterexample

Algebraic set identities

- We can use the laws of set identities given before to prove a statement of set theory
- Be extremely careful (even more careful than with propositional logic) to use the law exactly as stated

Algebraic set identity example

Theorem: $A - (A \cap B) = A - B$

Proof:

- $A - (A \cap B) = A \cap (A \cap B)^c$
- $= A \cap (A^c \cup B^c)$
- $= (A \cap A^c) \cup (A \cap B^c)$
- $= \emptyset \cup (A \cap B^c)$
- $= A \cap B^c$
- $= A - B$ ■

Prove or disprove

- For all sets A , B , and C , if $A \not\subseteq B$ and $B \not\subseteq C$, then $A \not\subseteq C$

Prove or disprove

- For all sets A and B , $\left((A^c \cup B^c) - A\right)^c = A$

Russell's Paradox

Naïve set theory

- Set theory is a slippery slope
- We are able to talk about very abstract concepts
 - $\{ x \in \mathbb{Z} \mid x \text{ is prime} \}$
- This is a well-defined set, even though there are an infinite number of primes and we don't know how to find the n^{th} prime number
- Without some careful rules, we can begin to define sets that are not well-defined

Barber Paradox

- Let a barber be the man in Westerville who shaves those men in Westerville who don't shave themselves.
 - Let T be the set of all men in Westerville
 - Let $B(x)$ be " x is a barber"
 - Let $S(x, y)$ be " x shaves y "
 - $\exists b \in T, \forall m \in T, B(b) \wedge (S(b, m) \leftrightarrow \sim S(m, m))$
- But who shaves the barber?



Russell's Paradox

- Bertrand Russell invented the Barber Paradox to explain to normal people a problem he had found in set theory
- Most sets are not elements of themselves
- So, it seems reasonable to create a set S that is the set of all sets that are not elements of themselves
- More formally, $S = \{ A \mid A \text{ is a set and } A \notin A \}$
- But is S an element of itself?

Escaping the paradox

- How do we make sure that this paradox cannot happen in set theory?
- We can make rules about what sets we allow in or not
- The rule that we use in class is that all sets must be subsets of a defined universe U
- Higher level set theory has a number of different frameworks for defining a useful universe

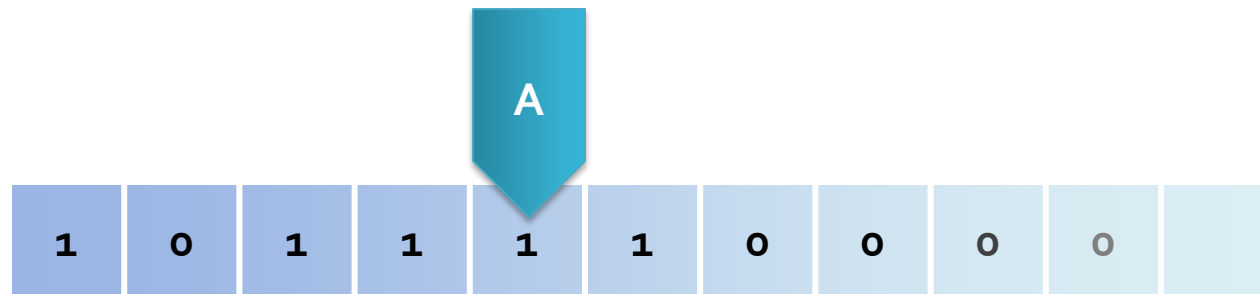
Halting Problem

Applying the idea again

- It turns out that the idea behind Russell's Paradox has practical implications
- It wasn't new, either
- Cantor had previously used a diagonal argument to show that there are more real numbers than rational numbers
- But, unexpectedly, Turing found an application of this idea for computing

Turing machine

- A Turing machine is a mathematical model for computation
- It consists of a head, an infinitely long tape, a set of possible states, and an alphabet of characters that can be written on the tape
- A list of rules saying what it should write and should it move left or right given the current symbol and state



Church-Turing thesis

- If an algorithm exists, a Turing machine can perform that algorithm
- In essence, a Turing machine is the most powerful model we have of computation
- Power, in this sense, means the ***ability*** to compute some function, **not** the ***speed*** associated with its computation

Halting problem

- Given a Turing machine and input x , does it reach the halt state?
- First, recognize that we can encode a Turing machine as input for another Turing machine
 - We just have to design a system to describe the rules, the states, etc.
- We want to design a Turing machine that can read another

Halting problem problems

- Imagine we have a Turing machine $H(m, x)$ that takes the description of another Turing machine m and its input x and returns 1 if m halts on input x and 0 otherwise
- Now, construct a machine $H'(m, x)$ that runs $H(m, x)$, but, if $H(m, x)$ gives 1, then $H'(m, x)$ infinitely loops, and if $H(m, x)$ gives 0, then $H'(m, x)$ returns 1
- Let's say that d is the description of $H'(m, x)$
- What happens when you run $H'(d, d)$?

Halting problem conclusion

- Clearly, a Turing machine that solves the halting problem **can't** exist
- Essentially, the problem of deciding if a problem is computable is itself uncomputable
- Therefore, there are some problems (called **undecidable**) for which there is no algorithm
- Not an algorithm that will take a long time, but **no algorithm**
- If we find such a problem, we are stuck
- ... unless someone can invent a more powerful model of computation

And it gets worse!

- Gödel used diagonalization again to prove that it is impossible to create a consistent set of axioms that can prove everything about the set of natural numbers
- As a consequence, you can create a system that is complete but not consistent
- Or you can create a system that is consistent but not complete
- Either way, there are principles in math in general that are true but impossible to prove, at least with any given system
- You might as well give up on math now

Ticket Out the Door

Upcoming

Next time...

- Cardinality
- Relations

Reminders

- Work on Assignment 4
 - Due next Friday
- Read 7.4 and 8.1
 - Write a three-sentence summary