META

Berlekamp-Welsh, Countability, Self Reference, Counting

1 General Comments

1. Berlekamp-Welsh:
   - Make sure students understand why the algorithm works
   - Suggestion is to have a quick walkthrough/review of Berlekamp-Welsh, and give the proof of why this algorithm works.
2. Countability:
   - #1-3
   - The set question
   - If A and B are both countable, then AxB is countable. (True/False)
3. Self Reference:
   - Take time to be very clear about the programs we are talking about: TestHalt/-Turing.
   - It might be helpful to draw out a flow chart to show how Turing(Turing) fails.
   - The key idea is that the existence of TestHalt is not possible (proof by contradiction), Turing generates this contradiction.
   - The equivalence between this and Cantor's diagonalization argument is again a bit tricky, try drawing it out in a table of (L)oop/(H)alt.
   - This relatively simple (if confusing) proof has huge implications. You might want to take some time to discuss what these are. It (like the related incompleteness

theorems by Gödel) but a limit on what we can know. We cannot generate arbitrary programs, and this gives a powerful tool (through the idea of reduction) to prove which programs are impossible.

4. Intro to Counting:
   - Mon-Wed people will not get to this
   - Poker
   - Solving Equations
5. Uncountability
   - Make sure that students are familiar with the common sets (integers, rationals, etc.)
   - The Hotel Argument
   - If students are confused about whether a set is countable or uncountable, ask them if the set can be enumerated in such a way that every element is covered.
   - Consider using proof by diagonalization (draw out the diagonalization) to explain uncountability for sets such as Infinite Bit String.
   - - Video
     - this provides a good mechanism for explaining the intuition behind finding a bijection depending on the level of preparedness of your students you may or may not have to draw out the hotels on the board
6. Self Reference
   - Turing program question is typically included in notes, but can be confusing. Definitely want to make sure students understand why this proof makes sense. Second part of that first question is good to understand but not central to the concepts of self-reference/computability. For question 2, suggest that students try to use the program to solve the halting problem (which if it can proves that the program cannot exist.
7. Counting
   - Definitely make sure to lecture on the rules of counting. Rather than just stating them, it will probably be better to do the examples with them in mind its a lot easier to understand them when you get why theyre used
   - Make sure to do the first one, as it emphasizes intuition of permutations vs. combinations, etc., and shouldnt take too long
   - Do the easier Starbucks counting ones to give them practice

- Solving equations-like stuff comes up decently often, so I definitely recommend going through that

2 Questions

2.1 Uncountability

1. **Cantor-Bernstein Theorem**

   - This theorem is not focused on (maybe not even really mentioned) in lecture, so the name and formal statement is just a neat thing to mention, BUT the technique that comes from it, namely the technique used to prove Qs countability, is very useful. Said technique states that if $|A| \leq |B|$ and $|B| \leq |A|$ then $|A| = |B|$.
   - You can do a proof of why Q is countable here.
   - We know that $|N| \leq |Q|$ because every natural number is a rational number.
   - Just need to show that $|Q| \leq |N|$.
   - Feel free to do a short version of the spiral proof. They will probably go over this in lecture, but if they didnt get it there you should cover it here. (Ask!)