Recitation 1 Solutions

Announcements

- Fill out the form about the time and level for your future recitations TODAY!
- Check the website for times and locations for all course events!
- Look at the course notes on the website for definitions and example proofs.
- Looking for a group? Use the teammate search on piazza. This week there are no GROUP problems, so you still have some time to find a group!
- The first homework assignment is out - start early!
- Recitations and solutions will be posted online. Please try to go through everything before attempting the solo problems on the homework! Extra problems are similar in difficulty to the problems we will cover, and provide more practice problems to go through. Bonus problems are a bit more difficult, but are still worth your time!

Common Proof Mistakes

As you prepare to do the writeups next week, remember the 10 styles of proof that you should avoid (adapted from Luis Von Ahn’s lecture notes):

- Proof by Stating Every Theorem in the Relevant Subject Area and Hoping for Partial Credit
- Proof by Example (“Here is a proof for $n = 2$. The general case is basically the same idea.”)
- Proof by Obscurity (using enough cumbersome and complex notation that your proof is impossible to decipher)
- Proof by Lengthiness (especially powerful combined with Proof by Obscurity and Proof by Stating Every Theorem)
- Proof by Switcheroo ($p \implies q$ is true, and $q$ is true, so $p$ must be true, right?)
- Proof by “It is clear that...” (“Clearly, this is the worst case for our algorithm.”)
- Proof by Generalization (This specific algorithm doesn’t work to solve this problem, and it seems optimal, therefore no algorithm can solve it! - a particularly common and pernicious flawed proof)
- Proof by Missing Cases (Your induction hypothesis doesn’t work for all cases, or your base cases are incomplete)
- Proof “by definition” (“By definition, our algorithm is $O(n)$”)
Some Definitions

- $\Sigma$ is your alphabet: non-empty and finite. Elements of $\Sigma$ are called symbols or characters.
- Given an alphabet $\Sigma$, a finite string or word over $\Sigma$ is a finite sequence of symbols, where each symbol is in $\Sigma$.
- $\Sigma^*$ is the set of all strings over $\Sigma$ of finite length, including the empty string $\epsilon$.
- Any subset $L \subseteq \Sigma^*$ is called a language over $\Sigma$.
- A computational problem is a function $f : \Sigma^* \rightarrow \Sigma^*$.
- A decision problem is a function $f : \Sigma^* \rightarrow \{0, 1\}$.
- There is a one-to-one correspondence between decision problems and languages.

Note that the proofs on this handout are bad or incorrect. Do not use them as a template.

Clearly false

Prove or disprove: for $n \in \mathbb{N}^+$, any $n$ people all have the same hair color.

Solution: We prove the claim via induction on $n$.

Base case ($n = 1$): trivial.

Induction hypothesis: Suppose that for some $n \in \mathbb{N}^+$, all groups of $n$ people have the same hair color.

Induction step: Consider a group of $n + 1$ people and take the entire group except one. By the IH, these $n$ have the same hair color. Now take another $n$-sized subgroup and exclude a different person. Again, these $n$ share a hair color. Now note that the two people who were excluded both have the same hair color as the $n - 1$ people who were picked twice - by transitivity, they must have the same hair color too. So all $n + 1$ people have the same hair color.

The induction step skims over the fact that when $n = 2$, the set of people picked both times is empty, which breaks the proof. Proofs of false things will generally score about 0 or 1 out of 10. When induction breaks, it usually breaks early on. So to sanity-check your induction proof, its a good idea to manually work through it for small cases like 1, 2, and 3. Also, lots of errors in proofs, even by professional mathematicians, can arise from trying to take an element of a set, without first proving the set is non-empty. Watch out for this.

Arrangement of ♣ and ♦

How many ways are there to arrange $c \geq 0$ ♣s and $d \geq 0$ ♦s so that all ♣s are consecutive?

Solution: You can have any number between 0 and $d$ ♦s, then a string of ♣s; then the remainder of the ♦s. Hence, there are $d + 1$ possibilities.

This proof ignores the edge case of $c = 0$, where there is only 1 possible arrangement. Edge cases are usually only worth a couple of points, so this proof would be in the 6-9 range. For some questions, the edge cases are the only interesting or hard cases, and so missing these cases would result in a very incomplete solution with a much lower score.
**Chips in a circle**

There is a circle of 15,251 chips, green on one side, red on the other. Initially, all show the green side. In one move, you may take any four consecutive chips and flip them. Is it possible to get all of the chips showing red?

**Solution:** No it is not possible. Let’s assume for contradiction we converted all 15,251 chips to red. But this means in the very last move there must be 4 consecutive green chips and the remaining 15,247 must be red. Repeating this \( k \) times for \( 1 \leq k \leq 3812 \), we get three consecutive red chips, with the rest green. But we started from all green, contradiction.

This solution tries to disprove the statement by assuming a particular process and then showing that the particular method won’t work. It does not suffice to show that one particular method does not work. It is unlikely that this kind of solution will score above 5.

Here is a correct proof of the statement:

We wish to show that it is not possible to reach a position of 15,251 red chips in a circle, beginning from the position of 15,251 green chips in a circle, in any number of moves, where a move consists of flipping the colors of four adjacent chips. To show this, we will first prove that if we make a move from any position with an even number of red chips (and 15,251 total chips) we will leave a position with an even number of red chips.

Note that whenever we flip a chip, we either increase or decrease the number of red chips by one, and therefore the parity of the total number of red chips changes (if it was odd, it becomes even, if it was even, it becomes odd). Since we can view each of the moves from the problem description as flipping four chips in succession, each move will cause the parity to change four times, meaning that the parity of the total number of red chips will end up remaining the same after each move. Thus, we have shown that if we make a move from a position with an even number of red chips, we will leave a position with an even number of red chips.

Note that the initial position has an even number of red chips, namely 0, and the target position has an odd number of red chips, namely 15,251. Assume for the sake of contradiction that there was some sequence of moves which began with the position with 0 red chips and ended with the position with 15,251 red chips. This sequence of moves begins at a position with an even number of red chips and ends at a position with an odd number of red chips, there must be some move in this sequence that goes from a position with an even number of red chips to a position with an odd number of red chips. But by the result we proved in the previous paragraph, any move from a position with an odd number of red chips leaves an odd number of red chips, so this is impossible. We have achieved a contradiction, and thus no such sequence of moves can exist.

**Balanced Parentheses**

Consider the following recursively defined language \( L \) over \( \{(),\}^* \):

- \((\) \in L,
- If \( x \in L \), then \((x) \in L\). (WRAP)
- If \( x, y \in L \), then \( xy \in L \). (CONCAT)
We claim that these rules create exactly “the set of balanced strings of parentheses”.
But what is that anyway?

(a) How do you reasonably define a “balanced string of parentheses”?

Either using the exact definition above, or the following:
Reading left to right, count +1 for ( and -1 for ), if your running total is always nonnegative and ends at zero, the string is balanced.

(b) Prove that a string $x$ over $\{(,\}\}^*$ is in $L$ if and only if it satisfies your above definition of a "balanced string of parentheses".
It may help to view this +1,-1 counter as a mountain range, where we start at the x axis, and repeatedly go up and down. The string is balanced then if we end at the x axis and never go below it. As an example, (())()() corresponds to:

```
/\ /\ /\ /\ /\ \ \\
```

It is sufficiently “obvious” that the counter definition corresponds immediately to the mountain range definition.

We wish to prove that \( L \) is exactly the set of strings that correspond to valid mountain ranges. We will do this with a double containment proof.

First, we prove that if a string is in \( L \) then it corresponds to a proper mountain range. We proceed by structural induction on the elements of \( L \).

Base Case: Note that () corresponds to /\, which is a valid mountain range (also the counter (+1,-1)), so the base case is complete.

Since there are two different rules for constructing new elements of \( L \) from existing ones, we have two cases to consider for our induction hypothesis and induction step.

First Induction Hypothesis: Suppose we have some string \( x \) which corresponds to a mountain range \( X \).

First Induction Step: We need to show that applying the WRAP rule corresponds to a valid mountain range. In particular, \((x)\) must correspond to a valid mountain range. We claim that we can do this by first putting a /, then the mountain range \( X \), then \. Visually, this is obvious, as since \( X \) is a valid mountain range, it starts and ends at the same height, so we can go up one at the beginning, down one at the end, never cross the x axis, and end up at the x axis. Additionally, this corresponds to our new string, as a / is +1 and \ is -1, our open and close parens. This completes this case of the induction.

Second Induction Hypothesis: Now, suppose we have a string \( x \) corresponding to range \( X \) and a string \( y \) corresponding to \( Y \).

Second Induction Step: We need to show that applying the CONCAT rule corresponds to a valid mountain range. This is easy to do. Just concatenate the \( X \) mountain range and the \( Y \) mountain range. This immediately corresponds to our new string of parentheses, and is clearly a valid mountain range, as both \( X \) and \( Y \) are by induction, concluding this direction of the proof.
Now, we need to show that every proper mountain range corresponds to a string in \( L \). Define the “height” of a mountain range to be the largest the counter ever gets, and the “length” to be the number of parenthesis characters that generated it.

We proceed by strong induction over the length of the range.

Base Case: Since we require a string of parentheses to be nonempty (to actually contain parentheses) and since a single parenthesis cannot be balanced, the smallest valid length for a string of balanced parentheses (or a mountain range) is 2. There is only one valid range of length 2, /\_. Since () is defined to be in \( L \), all length two mountain ranges correspond to strings in \( L \), concluding the base case.

Induction Hypothesis: Suppose that the claim holds for all mountain ranges of length less than \( n \), for some \( n \in \mathbb{N} \).

Induction Step: We have two cases, either the mountain range starts with a /, never touches the x axis until the end, then ends with \_\, or it touches the x axis between the beginning and end.

(a) In the first case, note that the “submountain” between the first and last character is a valid mountain range (starts height 1, ends height 1, never goes below by assumption). Since it is of length \( n - 2 \), by our induction hypothesis it corresponds to some \( x \in L \). Therefore by the WRAP rule, we have \( (x) \in L \) as desired, which is the parentheses string that exactly corresponds to our mountain range of length \( n \).

(b) In the second case, consider the mountain range up to the point where it hits the x axis. By splitting our range here, we have two strictly smaller valid mountain ranges (in terms of length), which correspond to some \( x, y \in L \) respectively by our IH. By CONCAT we have that \( xy \in L \), which corresponds to the full mountain range, concluding the proof.

**Inductio Ad Absurdum (Extra Problem)**

It is well known that \( \ln 2 \) is an irrational number that is equal to the infinite sum

\[
\sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} \ldots
\]

However, Leonhard claims to have a proof that shows otherwise:

He claims that \( \ln 2 \) is rational and will prove this by showing \( \sum_{i=1}^{n=1} \frac{(-1)^{i+1}}{i} \) is rational for all \( n > 0 \) via induction.

Base Case: \( n = 1 \): \( \sum_{i=1}^{1} \frac{(-1)^{i+1}}{i} = 1 \) is indeed rational.

Induction Hypothesis: Suppose that \( \sum_{i=1}^{n} \frac{(-1)^{i+1}}{i} \) is rational for \( 0 < n < k + 1 \) for some \( k \in \mathbb{N} \).

Induction Step: It now suffices to show that \( \sum_{i=1}^{k+1} \frac{(-1)^{i+1}}{i} \) is rational. We have that

\[
\sum_{i=1}^{k+1} \frac{(-1)^{i+1}}{i} = \sum_{i=1}^{k} \frac{(-1)^{i+1}}{i} + \frac{(-1)^{k+2}}{k+1}
\]

and by induction hypothesis, the first term is rational, and clearly the second term is also rational, and
since the sum of two rationals is rational, we are done!

Where did Leonhard go wrong?
While each step of his proof is correct, it doesn’t actually prove his claim! He only proved that the first $n$ terms of the summation is rational, but not that the entire summation is rational. In other words, he didn’t prove anything about the infinite sum.

**Induction (Extra Problem)**
Prove $n^2 \geq n$ for all integers $n$.

**Solution:** We prove $F_n = \frac{n^2}{n} \geq n$ by induction on $n$. The base case is $n = 0$: indeed, $0^2 \geq 0$. For the induction step, assume $F_k$ holds for all $k$. We now show that $F_{k+1}$ holds...

Write your induction hypothesis carefully: here we assume what we’re trying to prove (“for all $k$”). Because the IH is not even explicitly given, the meaning of $k$ is also somewhat ambiguous. Assuming the rest of the proof is correct, this kind of proof may score in the 5-8 range, depending on the difficulty of the actual induction step. If the induction step or base case is wrong (and it often is as a result of incorrectly set up induction), then this proof will probably score in the 0-3 range.

**Sneaky structures (Bonus Problem)**
Suppose that everyone in your recitation knows at least one other person in the recitation. We say that two students are connected if there exists a chain of students, each consecutive pair of which know each other, spanning between the two. For example, if Xavier knows Yvonne and Yvonne knows Zachary, then Xavier and Zachary are connected even if they don’t know each other. Prove or disprove: every student is connected to every other student.

**Solution:** We prove the claim via induction on $n$, the number of students.
Base case ($n = 2$): Since every student knows at least one other, the two students must know each other and are therefore connected.

Induction hypothesis: Suppose for some $k$ that this works for all groups of $k$ students. Induction step: Consider $k + 1$. We know the $k$-people recitation is connected. The $(k + 1)$th person cannot know no one, so they are connected to at least one other person in the recitation, who, by the induction hypothesis, is connected to everyone else. Thus, the whole party is still connected.

The implicit (false) assumption here is that the only way to construct graphs of minimum degree 1 is by adding one vertex at a time and connecting it to at least one previously existing vertex. (One counterexample: unions of disjoint graphs.) Correct graph induction is pretty tricky - if you’re curious about how it works, feel free to ask a TA.