Recitation - Introduction

1 Announcements

- The **first homework assignment is out**.
- General rule before starting the SOLO problems in the homework: First **carefully go through the course textbook** for definitions and example proofs. Second, go through recitation problems and solutions. (A homework problem can be a variation of a concept/proof covered in detail in the textbook or recitation.)
- *Extra* problems in the recitation handout are similar in difficulty to the other problems and provide more practice to go through. *Bonus* problems are a bit more difficult, but are still worth your time.
- Please fill out the recitation availability form as soon as possible!

2 Remember 76-101?

As you prepare to do the write-ups next week, keep in mind the following style guidelines:

- Writing good proofs requires as much attention to the principles of English composition as to those of mathematics.
- Apply your knowledge from 76-101 when you write proofs (as much as your knowledge from 21-127/21-128/15-151).
- Write in complete sentences and pay attention to grammar. Make sure your writing is organized in coherent sections.
- Avoid run-on sentences.

3 Some Definitions

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

4 Balanced Parentheses

Let Σ be the alphabet consisting of an opening parenthesis and a closing parenthesis, that is, $\Sigma = \{(,)\}$. Let language $L \subseteq \{(,)\}^*$ be inductively/recursively defined as follows.

- $() \in L$,
- 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 does that even mean?

1. Give a non-recursive definition of "balanced string of parentheses"?
2. Prove that a string x over $\{(,)\}^*$ is in L if and only if it satisfies your above definition of a "balanced string of parentheses".

Solution. https://www.youtube.com/embed/14pUrRCgMXQ

For part 1, a balanced string of parentheses is a non-empty string w in $\{(,)\}^*$ satisfying the following properties.

- **Prop 1**: The total number of open parentheses and closed parentheses in w are equal.
- **Prop 2**: When scanning w from left to right, at all points in time, the number of open parentheses is at least the number of closed parentheses.

We will denote by K the set of all strings satisfying **Prop 1** and **Prop 2**.

We now begin the proof for part 2. Our goal is to show $L = K$, and we will do so by a double containment argument. But before we begin, we introduce some notation that will allow us to express our proof more clearly and succinctly.[1]

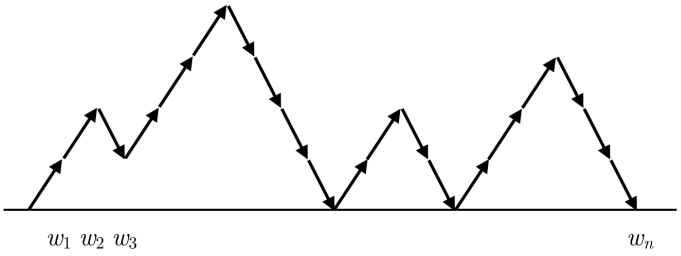
Given a string w in $\{(,)\}^*$, let $O_w(i)$ be the number of open parentheses in the substring $w[1 : i]$, which denotes the first i symbols of w. Similarly, let $C_w(i)$ be the number of closed parentheses in the substring $w[1 : i]$. Let $D_w(i)$ be $O_w(i) - C_w(i)$. And finally, let n be the length of w. Note that the O is for "open", the C is for "closed", and the D is for "difference". With this notation, we can express the two properties above as follows.

- **Prop 1**: $D_w(n) = 0$.

[1] This is an important skill to develop. Always look for opportunities to introduce notation that will clarify your proofs.

- **Prop 2**: For all $i \in \{1, 2, \ldots, n-1\}$, $D_w(i) \geq 0$.

One nice way to visualize these properties is with a "mountain range" picture:



As we scan the string from left to right, every time we encounter an opening parenthesis, we can think of it as an arrow going up, and every time we encounter a closing parenthesis, we can think of it as an arrow going down. The horizontal line represents height 0. We can think of $D_w(i)$ as the height after reading i symbols of the string. The first property says that at the very end, we should be back at height 0. The second property says that we never go below height 0. When these two properties are satisfied, we'll say that we have a *valid mountain range*. (Successfully solving problems is often about finding the right visualization!)

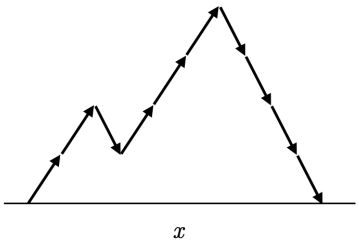
Part 1 : Showing $L \subseteq K$

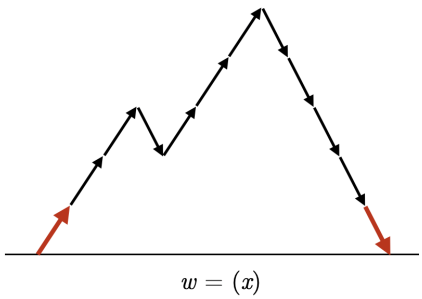
We will prove by structural induction[2] that for any $x \in L$, $x \in K$ as well.

Base Case. The base case corresponds to $w = \texttt{()}$ and it is easy to check that this w satisfies both **Prop 1** and **Prop 2**. Therefore $w \in K$.

Induction Step. Consider an arbitrary string $w \in L$ of length n such that $w \neq \texttt{()}$ (i.e. w does not correspond to the base case). This means that w was created using some $k \geq 1$ applications of the WRAP and CONCAT rules. Depending on what the last applied rule was, we know that either $w = \texttt{(}x\texttt{)}$ for some $x \in L$, or $w = xy$ for some $x, y \in L$. We consider both cases below. Note that the structural induction hypothesis allows us to assume that any string $x \in L$ that can be created using less than k applications of the WRAP and CONCAT rules, is in K.

- **Case 1**: $w = \texttt{(}x\texttt{)}$ where $x \in L$. By induction hypothesis, we know $x \in K$ and therefore satisfies **Prop 1** and **Prop 2**, i.e. x corresponds to a valid mountain range. Then w corresponds taking the mountain range x, lifting it by 1 unit of height, and attaching an up arrow at the beginning, and a down arrow at the end. This operation produces a valid mountain range.





Equivalently:[3]

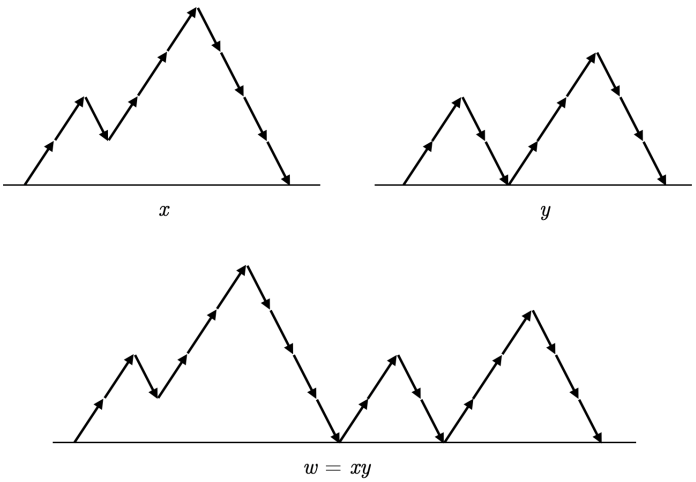
[2] Recall that structural induction is induction on the minimum number of applications of the recursive rules needed to construct a certain object.

[3] This portion is not really necessary and the argument using the mountain range already gives a satisfactory argument. However, it is a good exercise to spell things out using the formal notation introduced earlier.

- $O_w(n) = 1 + O_x(|x|)$, $C_w(n) = 1 + C_x(|x|)$, and therefore $D_w(n) = D_x(n) = 0$.
- $D_w(1) = 1$ and for every $i \in \{2, 3, \ldots, n-1\}$, $D_w(i) = 1 + D_x(i) > 0$.

These two observations show that w satisfies **Prop 1** and **Prop 2** respectively (i.e. that it is a valid mountain range). Therefore w is in K.

- **Case 2** : $w = xy$ where $x, y \in L$. By induction hypothesis, we know $x, y \in K$ and therefore both satisfy **Prop 1** and **Prop 2**, i.e. both are valid mountain ranges. The string w is obtained by concatenating two valid mountain ranges, so visually it is clear that w also corresponds to a valid mountain range.



Equivalently, let t be the length of x. Then observe that the following are true:

- $O_w(n) = O_x(|x|) + O_y(|y|)$, $C_w(n) = C_x(|x|) + C_y(|y|)$, and therefore $D_w(n) = D_x(|x|) + D_y(|y|) = 0 + 0 = 0$.
- For every $i \in \{1, 2, \ldots, t\}$, $D_w(i) = D_x(i) \geq 0$.
- $D_w(t) = D_x(t) = 0$, and therefore for $i \in \{t+1, \ldots, n\}$, $D_w(i) = D_y(i-t) \geq 0$.

The first item above shows that w satisfies **Prop 1**. The next two items show that w satisfies **Prop 2** (in fact the third item subsumes the first one, so we only really need the second and the third items). We conclude w is in K.

Part 2 : Showing $K \subseteq L$

We prove that if $w \in K$, then $w \in L$, by (strong) induction on the length of w.

Base Case. Note that K does not contain the empty string, so the shortest length w in K is the string `()`, which by definition is in L. There are no other strings of length 2 that are in K.

Induction Step. Let w be an arbitrary string in K of length $n > 2$. The (implicit) induction hypothesis is that any string $w' \in K$ of length less than n is in L. By the definition of K, we know w satisfies **Prop 1** and **Prop 2**, i.e., it is a valid mountain range. We will consider two cases.

- **Case 1**: There exists $t \in \{2, \ldots, n-1\}$ such that $D_w(t) = 0$.

  Define $x = w[1:t]$ and $y = w[t+1:n]$ so that $w = xy$. Then we claim that both x and y satisfy **Prop 1** and **Prop 2**. In other words, both x and y are valid mountain ranges. Visually this is not hard to see. Or equivalently:

  - For $i \in \{1, 2, \ldots, t\}$, $D_x(i) = D_w(i)$. Therefore $D_x(t) = D_w(t) = 0$ and $D_x(i) = D_w(i) \geq 0$ for $i \in \{1, 2, \ldots, t-1\}$.

- For $i \in \{1, 2, \ldots, n-t\}$, $D_y(i) = D_w(t+i)$. Therefore $D_y(n-t) = D_w(n) = 0$ and for $i \in \{1, 2, \ldots, n-t-1\}$, $D_y(i) = D_w(t+i) \geq 0$.

Since x and y satisfy **Prop 1** and **Prop 2**, they both belong to K. And by induction hypothesis, they both belong to L. This implies that $w = xy$ is also in L as it can be constructed from two strings in L using the CONCAT rule.

- **Case 2** : There does not exist $t \in \{2, \ldots, n-1\}$ such that $D_w(t) = 0$. In other words, for all $t \in \{2, \ldots, n-1\}$, $D_w(t) \geq 1$. Let's mark this statement with $[*]$.

  We know that w must start with an opening parenthesis. And since $D_w(n) = 0$, we also know that w must end with closing parenthesis (otherwise we would have $D_w(n-1) = -1$). Therefore we can write w as (x) for some string x of length $n-2$. We claim that x satisfies **Prop 1** and **Prop 2**, i.e. x is a valid mountain range. This follows from $[*]$.

  - $D_x(n-2) = O_x(n-2) - C_x(n-2) = (O_w(n) - 1) - (C_w(n) - 1) = D_w(n) = 0$.
  - For $i \in \{1, 2, \ldots, n-2\}$, $D_x(i) = D_w(i+1) - 1 \geq 1 - 1 = 0$, where in the last inequality, we used $[*]$ above.

  Since x satisfies **Prop 1** and **Prop 2**, it belongs to K. And by induction hypothesis, it belongs to L. This implies that $w = (x)$ is also in L as it can be constructed from a string in L using the WRAP rule.

In both cases, we were able to show that $w \in L$, as desired. ■

Note that the solutions/proofs below are bad or incorrect. Do not use them as a template.

5 Clearly false

Prove or disprove: Any n points on the plane are collinear (they are all in the same line).

Proposed answer: We prove the claim via induction on n.

Base case ($n \in \{1, 2\}$): trivial.

Now let $k > 2$ be some arbitrary natural number. Assume the statement holds for all sets of $k-1$ points on the plane. We want to show that it also holds for any k points. So consider any k points on the plane. By induction hypothesis the first $k-1$ points are one line, and also the last $k-1$ points are on a line. Therefore, all the k points are on a line.

Critique this answer.

Solution. https://www.youtube.com/embed/NYCx7yi37M8

The proof breaks when we try to establish the claim for $k = 3$. To prove that the union of two sets of collinear points is collinear, we need the two sets to intersect in at least two points. However, we do not have that when we try to prove the $k = 3$ case.

When induction breaks, it usually breaks early on. So to sanity-check your induction proof, it is a good idea to manually work through it for small cases like $n = 1$, $n = 2$, and $n = 3$. 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. ■

6 Arrangement of clubs and diamonds

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

Proposed answer: 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.

Critique this answer.

Solution. https://www.youtube.com/embed/I_9j1m07dAw

This proof ignores the edge case of $c = 0$, where there is only 1 possible arrangement. Mistakes in edge cases are usually considered to be minor. However for some problems, the edge cases are the only interesting or hard cases. Missing these cases would result in an unsatisfactory grade. ■

7 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?

Proposed answer: 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.

Critique this answer.

Solution. https://www.youtube.com/embed/jPPpGiWXXu4

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.

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. By the property we just proved, no matter how many moves we make, the invariant of having an even number of red chips is preserved. Since 15,251 is an odd number, we can conclude that we can never reach a position with 15,251 red chips. ■

8 Inductio Ad Absurdum (Extra Problem)

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

$$\sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \dots$$

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

He claims that $\ln 2$ is rational and will prove this by showing $\sum_{i=1}^{n} \frac{(-1)^{i+1}}{i}$ 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?

Solution. https://www.youtube.com/embed/0rXtL4HGOp0

While each step of his proof is correct, it doesn't actually prove his claim! He only proved that for all $n \in \mathbb{N}$, 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. ■

9 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.

Proposed answer: 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 $k+1$ students. We know the k-student 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.

Critique this answer.

Solution. https://www.youtube.com/embed/wA09I0do2bo

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 requires care - if you're curious about how it works, feel free to ask a TA. The proper way to induct on graphs will be taught later in this course! ■