CS 30: Discrete Math in CS (Winter 2019): Lecture 1

Date: 3rd January, 2019 (X-hour)
Topic: Discrete Math: What, Why, How?; Direct Proofs
Disclaimer: These notes have not gone through scrutiny and in all probability contain errors.
Please discuss in Piazza/email errors to deeparnab@dartmouth.edu

1 Discrete Math: what and why?

Mathematics forms the bedrock of all sciences and computer science is no exception. The difference, however, from the other natural sciences is in the type of mathematics. In the natural sciences, one needs the language of mathematics to describe things in *continuous* motion – think electricity, reagents, cell fluids, prices, etc. In computer science, we need the language of mathematics to argue about *discrete* objects such as bits, integers, a set of files, a network, etc. The first objective of this course is to teach you that language; a language you will be using in all other aspects of computer science.

The second objective of this course is to focus on *proofs*. In this course, we will make many *precise, rigorous* statements and prove them. We will learn various styles of proving, and we will see many examples of proofs. But there is only one way to really learn how to prove, and that is also the same way one learns how to be a good woodworker – hours and hours of practice; you will clock in a few before we are done.

2 Direct Proofs

- **First proof of CS 30.** We begin the class with a simple fact.

  Theorem 1. The sum of two odd numbers is even.

  The first thing when faced with a proposition is to *verify* the truth by looking at some examples. Here goes: $3 + 3 = 6$, and 6 is even. Good. $1 + 7 = 8$ and 8 is even. Good. One could stop here but that won't be a proof. That would be nothing more and nothing less than looking at some examples. Here is the proof.

  Proof. Let a and b be *any* two odd numbers. Since they are odd, $a = 2k + 1$ for some number k and $b = 2\ell + 1$ for some number ℓ. These two numbers k and ℓ are completely defined by a and b (for instance, they can be found by doing $a//2$ and $b//2$, respectively, in Python 3.) Therefore,

  $$\begin{aligned} a + b &= (2k+1) + (2\ell+1) \\ &= 2k + 2\ell + 2 \\ &= 2(k + \ell + 1) \end{aligned}$$

  That is, $a + b$ is even. □

✍

Exercise: *Prove : The* product *of any two positive odd numbers is always odd.*

✍

Exercise: *Prove : If a is a positive* even *integer, then a^2 is divisible by* 4.

✍

Exercise: *Prove that the sum, difference, and product of two rational numbers is always rational. Furthermore, if both are non-zero, then their ratio is also rational.*

- **The Summation Formula.** The summation operator is *very much* like the $\int$ operator/notation from calculus. Consider a *sequence* of numbers $a_1, a_2, \ldots$'s which are indexed by i. The variable i is the *iterator* which is *usually* an *integer* and starts at p and initiates a counter (called sum) to 0. If $q \geq p$, then the iterator increments by 1 till it reaches q adding a_i to the counter. If $q \leq p$, then the iterator decrements by 1 till it reaches q adding a_i to the counter. The answer is the final value of the counter sum. In short, $\sum_{i=p}^{q} a_i$ is the output of the following for-loop; in particular, for any summation given to you, you can write the following code and find the answer.

$\sum_{i=p}^{q} a_i$:=

```
sum = $0$.
for $i = p$ to $q$ do:
    ▷ If $q \geq p$, $i$ is incremented; otherwise, decremented.
    sum = sum + $a_i$.
return sum.
```

- **Sum of the first n natural numbers.**

Theorem 2. For any number $n \geq 1$, we have

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

As before, you should check if the claim is correct. For instance, if I set $n = 3$, the LHS evaluates to $1 + 2 + 3 = 6$ and the RHS evaluates to $3 \cdot 4/2 = 6$. Another sanity check you should do is that the LHS is an integer, is the RHS an integer? Yes it is – for *any* n either n or $n + 1$ is even, and this divisible by 3. ✍

Exercise: *Is $\frac{n(n+1)(2n+1)}{6}$ always an integer for any $n \geq 1$? You may wait a couple of weeks to do this exercise.*

Proof. We learn an important trick of *change of variables*. Note that changing variables $j = n - i + 1$, we can write we can write

$$S = \sum_{i=1}^{n} i = \sum_{j=n}^{1} (n - j + 1) = \sum_{j=1}^{n} (n - j + 1) = \sum_{i=1}^{n} (n - i + 1)$$

The first equality is definition. In the second equality, we have changed the variables $j \leftarrow n-i+1$. The third equality follows from *commutativity of addition*. The fourth equality notes that the name j can be changed back to i. Therefore, we get

$$2S = \sum_{i=1}^{n} i + \sum_{i=1}^{n} (n-i+1) = \sum_{i=1}^{n} \Big(i + (n-i+1) \Big) = (n+1) \cdot \sum_{i=1}^{n} 1 = (n+1) \cdot n$$

□

Exercise: *Is the above formula correct for $n < 1$? How would the formula change?*

Exercise: *Given two integers $a \leq b$, write a formula for $\sum_{i=a}^{b} i$. Your final answer should only depend on a and b (and not i).*

Exercise: *An arithmetic progression (AP) is defined by two integers (a, d). Given these, the corresponding AP is the sequence $a_1, a_2, a_3, \ldots$ where $a_i := a + (i-1) \cdot d$. For example, the sequence $1, 2, 3, \ldots$ is the AP generated by $(1, 1)$. Similarly, $2, 4, 6, 8, \cdots$ is the AP generated by $(2, 2)$.*

Given (a, d) and the corresponding AP sequence $a_1, a_2, \ldots$, write a formula for $\sum_{i=1}^{n} a_i$. Your answer should depend only on a, d, and n.

Once you have proved and know the above theorem, you can prove more theorems as *corollaries*.

Corollary 1. The sum of the first n *even* numbers is $n(n+1)$.

Proof. The ith even number is $2i$. Therefore, the sum of the first n *even* numbers is

$$D := \sum_{i=1}^{n} (2i)$$

But the RHS is nothing but

$$\sum_{i=1}^{n} (2i) = 2 \left(\sum_{i=1}^{n} i \right)$$

And the paranthesized term in the RHS is what we figured out in Theorem 2. Substituting, we get $\sum_{i=1}^{n} (2i) = 2 \cdot \left(\frac{n(n+1)}{2} \right) = n(n+1)$, which is what we set out to show. □

Exercise: *What is the sum of the first n odd numbers?*

Exercise: *As mentioned in class, the above "trick" doesn't work for summing the first n* squares. *Here is a useful exercise: mimic the proof of Theorem 2 and figure out where you get stuck.*

Remark: *Before we move on, let me give a (famous) example of why* proofs by examples *are not a great idea. Consider the following proposition.*

Proposition 1. (Beware!) The number $n^2 + n + 41$ is a prime for all number $n \geq 1$.

Let's try $n = 1$; we get $n^2 + n + 41 = 43$ which is prime. Good. Let's try $n = 2$; we get $n^2 + n + 41 = 47$ which is prime. Good. Let's try $n = 3$; we get 53 which is prime. Good! Ok, so let's try a random n – let's try $n = 10$; we get $n^2 + n + 41 = 151$ which is prime! Ok, this must be true.

It is quite tempting to conjecture *at this point that the above proposition may be true. Indeed, if we had no computers and were doing this by hand, we would get primes all the way to $n = 40$. On the other hand, $41^2 + 41 + 41$ is, well, 41×43. So, no, the proposition isn't true. It had a good run, but then it fell. We found a* counterexample *to the proposition, thereby proving it is not true! All it takes is one counterexample.*

But there are many other propositions for which neither has one found a proof, nor has one found a counterexample. Here is another famous one (there are many involving numbers):

Proposition 2. Every even number greater than 4 can be written as a sum of two primes.

You can, and you should, start checking. And many have. A lot. All the way up to 400 trillion. No counterexample. But no proof yet either. The world is waiting...