Fall, 2013 CIT 592

I wanted to write up these problems in a more dialogue style since if I was teaching this, we would have gone through these steps of logical reasoning. So yes, you will see steps being taken over here that are basically going to be crossed out. I want to show them to you because this is likely to be the way that you think.

After a while of course, you get familiar enough with induction to say, let's just do strong induction. Or wait, this is just weak induction, since we are dealing with sequences. Or wait, this is some kind of combinatorics question, maybe weak induction works the moment I take my nth item aside and examine the remaining.

Problems.

- Show inductively how 4-cent stamps and 5-cent stamps are sufficient to get postage for anything greater than or equal to 12 cents

  Proof by induction

  Base case : - Can we generate 12 cents worth of postage with just 4-cent stamps and 5-cent stamps? Yes. Take 3 4-cent stamps and you have 12 cents worth of postage.

  Assume inductively that for $n \in \mathbb{N}, n > 12$, $n - 1$ cent postage can be generated by using just 4-cent and 5-cent stamps.

  To prove:- If $n-1$ cent postage can be generated, then n cent postage can be generated too.

  We stare at the above statement for a while and realize that from $n - 1$ cent postage, there is really no indication for how to make n cent postage.

  Consider a concrete example of the failure of this. I showed how I can do 12 cent postage. But how do I make 13 cent postage. Well, it is 2 4-cent stamps and 1 5-cent stamp. Did I get to that realization by looking at the 12 cent postage? Clearly not. In this case, I just used 'common' sense.

  Bottom line, weak induction is not helping me!!

  Let us try strong induction

  $$\text{Assume inductively that } \forall k \in \mathbb{N}, 12 \leq k < n \text{ we can generate k cent postage.} \quad (1)$$

  Now to show how to generate postage for n.

  This part is admittedly a tiny bit tricky. The key insight is if only I knew how to produce postage for $n - 4$, then I know how to produce postage for n. After all, I can take my method for generating postage for $n-4$ and then just add in one more 4-cent stamp.

Do I know how to generate postage for $n-4$? Statement (1) above seems to suggest I do. But we need to be careful, I can only generate postage if $12 \leq n-4 < n$. Well $n-4$ is definitely less than n. But $n-4$ will exceed 12 only in situations when $n > 15$.

So this tells me that my inductive proof works for $n > 15$. For any $n > 15$, I use the fact that the inductive hypothesis gives me a way to generate postage for $n-4$. Adding in a 4-cent postage stamp, I can readily produce postage for n.

But what about n= 13,14,15. Those are additional base cases! So this problem reveals another different aspect of strong induction. In strong induction, on occasion, you will have to do more work for base cases.

So we now have to show how postage is generated for 13, 14, 15.

$$13 = 4 \times 2 + 5$$
$$14 = 4 + 5 \times 2$$
$$15 = 5 \times 3$$

The above three equations show me the postage generation for 13,14,15.

At this point I claim I am done by strong induction.

To review, we have shown how to generate postage for 12,13,14,15. We have then shown using strong induction that if know how to generate postage for any $12 \leq k < n$, then we can generate postage for n (specifically in this case, what we needed was n-4. Note that different strong induction proofs will make use of different parts of the induction hypothesis).

Read this page if you are dying to know how this ties in with CS programming.

Imagine a CS programming question that says write a program to show how to generate postage greater than 12-cents using just 4 and 5 cent stamps.

This is what the function looks like in what algorithms guys will call pseudocode

```
postageGenerator(n){
        if 12 return 3 4s and 0 5s
        if 13 return 2 4s and 1 5
        if 14 return 1 4 and 2 5s
        if 15 return 0 4s and 3 5s
        if bigger than 15 return postageGenerator(n-4) + 1 more 4
}
```

A function calling itself? What?? Well, if you've not covered it yet, you will. This is recursion! And in my personal opinion, the induction - recursion connection is one of the biggest reasons to see discrete math in a CS curriculum.

So thinking about the problem through the perspective of how do I solve it via induction actually helps me write the program!

Here is what it looks like in Python, just to show it in some concrete language. I think even if you have not seen Python syntax, it is one of the more understandable languages, so hopefully this is not beyond your understanding.

```
def postageChangeGenerator(n):
    l = []
    if n==12:
        return [3,0]
    elif n == 13:
        return [2,1]
    elif n == 14:
        return [1,2]
    elif n == 15:
        return[0,3]
    else:
        return [postageChangeGenerator(n-4)[0] + 1,
                postageChangeGenerator(n-4)[1]]
```