BIL 133 HW3

Murat Sahin
191101010

June 20, 2020

1 PROVING HUMMINGBIRDS ARE SMALL

1.1 Explaining reasoning informally

(P1) "All hummingbirds are richly colored."
(P2) "No large birds live on honey."
(P3) "Birds that do not live on honey are dull in color."
(C1) "Hummingbirds are small."

Lets merge last two premises, we will conclude that large birds are dull in color. Because second premise says large birds does not live on honey and third premise says if a bird does not live on honey it means it is dull in color.

We found that large birds are dull in color. If we look at first premise, it says "All hummingbirds are richly colored.". So if they are richly colored, they can not be large. Thus, hummingbirds are small.

1.2 Stating expressions

P(x) = "x is a hummingbird"
Q(x) = "x is large"
R(x) = "x lives on honey"
S(x) = "x is richly colored"

$$\text{(P1)} = \forall x(P(x) \rightarrow S(x))$$
$$\text{(P2)} = \forall x(Q(x) \rightarrow \neg R(x))$$
$$\text{(P3)} = \forall x(\neg R(x) \rightarrow \neg S(x))$$
$$\text{(C1)} = \forall x(P(x) \rightarrow \neg Q(x))$$

1.3 Proving the conclusion

$\forall x(P(x) \to S(x)), \forall x(Q(x) \to \neg R(x)), \forall x(\neg R(x) \to \neg S(x)) \vdash \forall x(P(x) \to \neg Q(x))$

1	$\forall x(P(x) \to S(x))$	Premise
2	$\forall x(Q(x) \to \neg R(x))$	Premise
3	$\forall x(\neg R(x) \to \neg S(x))$	Premise
4	$P(x_0) \to S(X_0)$ $\quad x_0$	$\forall x_e$ 1
5	$Q(x_0) \to \neg R(x_0)$	$\forall x_e$ 2
6	$\neg R(x_0) \to \neg S(x_0)$	$\forall x_e$ 3
7	$P(x_0)$	Assumption
8	$S(x_0)$	$\to_e$ $7, 4$
9	$\neg\neg S(x_0)$	$\neg\neg_i$ 8
10	$\neg\neg R(x_0)$	Modus Tollens 6,9
11	$\neg Q(x_0)$	Modus Tollens 5,10
12	$P(x_0) \to \neg Q(x_0)$	$\to_i$ $7 - 11$
13	$\forall x(P(x) \to \neg Q(x))$	$\forall x_i$ $4 - 12$

2 UPPERBOUND FOR A VARIATION OF TOWER OF HANOI

First, we need to find S(0), S(1) and S(2)

S(0) = 0 since there is no need to transfer any disks.
S(1) = 1 we only need to transfer only one disk to another peg.
S(2) = 3 we will transfer top disk to any different peg, then we will transfer other disk to any empty peg. Now, big disk is in correct position, as a last move we will transfer small disk on the big disk.

We will solve this problem by splitting n disks into two groups. k is an integer, one group will be smallest k disks, other will be the rest.

1. Transfer smallest k disks to another peg using all four pegs. This can be done using function S because group of large disks does not block their moves. With that move, we split groups between two pegs. (S(k) moves)
2. Transfer all disks in group of large disks except largest to another **empty** peg using all pegs except the peg that group of small disks stand. This can be done using function T because largest disk does not block their moves. Now, there is one empty peg left, this will be our final peg. (T(n-k-1) moves)
3. Transfer biggest disk to empty peg. (1 move)
4. Transfer all disks in group of large disks except largest to peg that biggest disk stands using all pegs except the peg that group of small disks stand. This can be done using function T because largest disk does not block their moves. (T(n-k-1) moves)
5. Transfer smallest k disks to peg that biggest disk stands using all four pegs. This can be done using function S because group of large disks does not block their moves. (S(k) moves)

We transferred n disks from one peg to another while obeying Lucas' rules.

Well, great. But did we transferred them in the best way?

We can not say

$$S(n) = 2S(k) + 2T(n-k-1) + 1$$

because we did not proved that our solution is the optimal one. We can say:

$$S(n) \leq 2S(k) + 2T(n-k-1) + 1 \text{ for any } k \in \{1, 2, ..., n-1\}.$$

3 INDUCTION PROOF FOR GEOMETRY

Define P(m) : all m-sided polygons' sum of internal angles is $(m-2)\pi$ for $m \geq 3$.

Claim : P(m) is true for all integer m that holds $m \geq 3$.

Proof : We will prove this claim by mathematical induction on m.

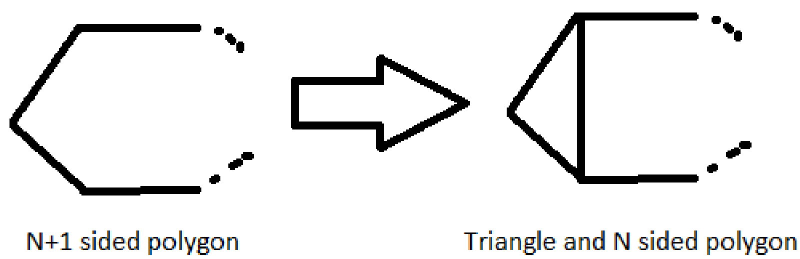
Base Case : As the base case, we consider the case where m=3.

$(3-2)\pi = \pi$ So, P(3) holds from our calculation and given premise.

Inductive Hypothesis : The claim is true for some integer n that holds $n \geq 3$.

Inductive Step : All we need to do is show that P(n+1) is true.

Lets think n+1 sided polygon. If we split it into two pieces like below, we will get one triangle and one n sided polygon.



Sum of the interior angles in n+1 sided polygon is equal to sum of the interior angles in n sided polygon plus sum of the interior angles of a triangle. P(3) is true from premise and P(n) is true from our inductive hypothesis. With that information, we can find sum of the interior angles in a n+1-sided polygon.

$P(3) = \pi$ from our premise.
$P(n) = (n-2)\pi$ since P(n) is true from our inductive hypothesis.

$$P(n+1) = P(3) + P(n)$$
$$P(n+1) = \pi + (n-2)\pi$$
$$P(n+1) = (n-1)\pi$$

If our calculation is equal to $(m-2)\pi$ for m=n+1, P(n+1) is true.

$$(n-1)\pi \stackrel{?}{=} (n+1-2)\pi$$
$$(n-1)\pi = (n-1)\pi$$

So, P(n+1) is true.
Hence, our proof by mathematical induction is completed. P(n) is true for all $n \geq 3$.
So, sum of internal angles in a n-sided polygon (where $n \geq 3$) is equal to $(n-2)\pi$.

4 ANOTHER VARIATION OF TOWER OF HANOI

- Lets look at the small cases.
  For n = 0 there can not be a move since there are no disks. 0 moves will be necessary.
  For n = 1 we have two disks with the same size. We will carry top disk to another peg then we will carry other disk to the same peg. 2 moves will be necessary.
  For n = 2 we have four disks with the same size. To put largest two disk to another peg, we need to carry smallest two disk. We will carry smallest disks one by one to another peg in 2 moves. Then, we will carry biggest disks one by one to empty peg in 2 moves. Finally we will carry smallest disks one by one back on the biggest disks in 2 moves. 6 moves will be necessary.

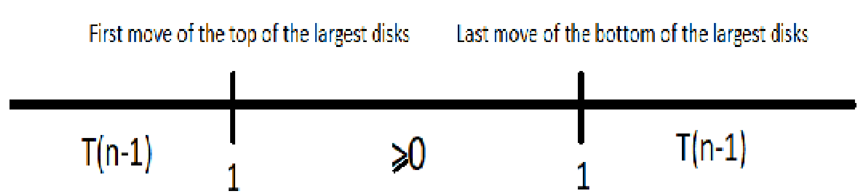
  It looks like the classic Tower of Hanoi with double moves.

- Let T(n) be the minimum number of moves necessary and sufficient to transfer 2n disks of n different sizes, two of each size, from a peg to another peg while obeying updated Lucas' rules.

- **1)** First, transfer smallest 2n-2 disks to a different peg in T(n-1) moves. Largest disks does not affect usage since every disk can be carried onto them.
  2) Then, transfer largest disks to empty peg one by one in 2 moves.
  3) Finally, transfer smallest 2n-2 disks back onto the largest in T(n-1) moves. Largest disks does not affect usage since every disk can be carried onto them.

$$T(n) \leq 2T(n-1) + 2 \text{ for all } n > 0.$$

- Optimal algorithm should carry biggest two disks to their final peg. They need to move one by one, so lets start with top one. In order to carry it, other disks other than its twin should be stacked in a different peg. I marked its first movement on the time-line. Then, lets consider the other largest disk's last movement. It must go onto its twin in last movement. If it is the last movement, our job is done with the largest disks. Finally, we can carry smallest 2n-2 disks onto our largest disks. Check the time-line below.



So, $T(n) \geq 2T(n-1) + 2$ for all $n > 0$.

- With combining upper and lower bounds, we can obtain a recurrence equality below.

$$T(n) = 2T(n-1) + 2 \text{ for all } n > 0.$$

- Now, lets solve our recurrence.

$$T(n) + 2 = 2T(n-1) + 4 \text{ for all } n > 0.$$
$$\text{Let } U(n) = T(n) + 2 \text{ for all } n > 0.$$
$$U(0) = T(0) + 2 = 0 + 2 = 2$$
$$U(n) = 2U(n-1) \text{ for all } n > 0.$$

$$\text{Then, } U(n) = 2^{(n+1)}$$

$$\text{Then, } T(n) = 2^{(n+1)} - 2$$

This is our general solution for this problem.

5 NUMBER OF BINARY STRINGS IN THE SPECIAL FORM

Lets look at the small cases.

For n=0, F(0) = 1 since empty string does not have 2 consecutive 1's in it.
For n=1, F(1) = 2 since both strings "0" and "1" does not have 2 consecutive 1's in it.
For n=2, F(2) = 3 since the strings "00","01" and "10" does not have 2 consecutive 1's in it.
For n=3, F(3) = 5 since the strings "000","100","010","001","101" does not have 2 consecutive 1's in it.

Let $n \geq 2$
.
Define S(n) as the set of the binary strings of length n with no two consecutive 1's in it.

How can we find elements of S(n)?

Binary strings, if not empty, end with either "1" or "0". Lets find them one by one and add them up.

To produce S(n)'s elements that end with "0", we can append elements of S(n-1) by adding "0" to the end. We don't need to check it is valid or not because adding "0" to the end can not generate 2 consecutive 1's. Moreover, if we undo the process, we are able to get all of the S(n-1)'s elements. So, number of S(n)'s element that end with "0" is F(n-1).

To produce S(n)'s elements that end with "1", we can not just append elements of S(n-1) by adding "1" to the end. It may generate 2 consecutive 1's. In order to prevent that, we must take elements of S(n-1) which ends with "0". As we showed above, elements of S(n-1) which ends with "0" produced with adding "0" to end of the S(n-2)'s elements. Thus, we can add "01" to end of the S(n-2)'s elements. Moreover, if we undo the process, we are able to get all of the S(n-2)'s elements. So, number of S(n)'s element that end with "1" is F(n-2).

We found them, lets add them up and get a recurrence relation.

$F(1) = 2$, $F(2) = 3$ and $F(n) = F(n-1) + F(n-2)$ for all $n \geq 2$