CS70–Fall 2011 — Solutions to Homework 3

October 14, 2016

1. Strengthening the Proposition

   Claim: For all $n \geq 1$, all entries of the matrix $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^n$ are bounded above by n.

   Proof: by Induction

   Let $P(n)$ be the proposition $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^n$. In order to strengthen the proposition, we'll evaluate $P(n)$ for few values of n.

   For $n = 1$
   $$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

   For $n = 2$
   $$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$$

   For $n = 3$
   $$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^3 = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}$$

   A pattern arises as we evaluate $P(n)$ for various values of n. Therefore, we can write the above proposition as,

   $$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^n = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix}$$

   If we are able to prove this, then we have also proved the claim that $\forall n \in N$ all the entries of $P(n)$ are bounded above by n.

   Base Case: For $n = 1$, we have

   $$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \checkmark$$

Inductive Step: Suppose that $P(n)$ is true. Therefore $P(n+1)$ becomes

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{n+1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{n} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

As we know that $P(n)$ is true. Therefore,

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{n+1} = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ n+1 & 1 \end{pmatrix} \checkmark$$

2. Principle of Induction

(a) $(\forall n \in \mathbb{N})(P(n))$.
Possibly True. In order for the proposition $P(n)$ to be true, there must be a base case. As there is no base case here, it is not possible for us to say that's its true. Also there is no reason for us to say that its false.

(b) $(\forall n \in N)(\neg P(n))$.
Possibly True. In order for the proposition $\neg P(n)$ to be true, there must be a base case. As there is no base case here, it is not possible for us to say that's its true. Also there is no reason for us to say that its false.

(c) $P(0) \implies (\forall n \in N)(P(n+2))$.
Possibly True. This statement is also partially true. The above proposition is true for even numbers. However, it says nothing about odd numbers whatsoever. Therefore, we can't make any conclusion.

(d) $(P(0) \land P(1)) \implies (\forall n \in N)(P(n))$.
Definitely True. The first base case $P(0)$ shows that the proposition is true for all even numbers, while the second base case $P(1)$ shows that the proposition is true for all odd numbers. Because the proposition is true for all even numbers and all odd numbers, we can say that the proposition is true $\forall n \in N$.

(e) $(\forall n \in N)P(n) \implies ((\exists m \in N)(m > n + 2011 \land P(m)))$.
Definitely True. As given in the problem statement, we know

that $P(n) \implies P(k)\ \forall k > n$ is true, if k has the same parity(evenness or oddness) as n. Here, for any given value of n, there is always a $m > n+2011$ with the same parity as n. Therefore, the statement is true.

(f) $(\forall n \in N)(n < 2011 \implies P(n)) \wedge (\forall n \in N)(n \geq 2011 \implies \neg P(n))$.

Definitely False. Consider only the first half of the proposition, then we have base cases for both parities(evenness and oddness), and thus that part of the proposition is true. However, this contradicts the latter half of this proposition, and we know that $T \wedge F$ is $false$. Therefore, this statement is always false.

3. The Proof of the Pi is in the Eating

Claim: Dave can always, via some sequence of moves, arrange the pizzas in order of size.

Proof: This is possible.

(a) Search for the largest pizza in the stack.

(b) Place the spatula under this pizza and flip. After this, you will have the largest pizza at the top of stack.

(c) Place the spatula under the bottom pizza and flip. After this, you will have the largest pizza box at the bottom of the stack.

(d) Do the same process for the second largest pizza, but instead of flipping from the bottom pizza, you flip from the pizza above it.

(e) Repeat this process for n iterations in order to sort n pizza's.