Topics: Bad Proofs, Stable Marriage, Cake Cutting

1 Bad Proofs

Consider the following proof:

Claim: Every natural number can be described in fifteen English words or less.
Proof by well-ordering.
Let $P(n)$ = "n can be described in fifteen English words or less".
Let us assume that there are numbers n for which $P(n)$ is not true. In this case, by the well-ordering principle, there must be a least such number, and let that number be k. But then k can be described by the following sentence "It's the smallest natural number that *cannot* be described in fifteen English words or less." However, this is a description that requires fifteen words or less, and so we have a contradiction. Thus $P(n)$ is true for all numbers n.

What, if anything, is wrong with the above proof? The claim itself is easy to disprove. Let N be the number of words in the English language. Then there are about N^{15} possible descriptions of 15 words or less. Thus if the claim were true, there would be at most N^{15} natural numbers. We know this is not the case, so the claim is false.

The problem in the proof is that $P(n)$ is actually not a well-defined proposition. In fact, being "describable in fifteen English words or less" depends not only on the encoding of the numbers, but also on the context in which this is spoken, and to define it, we need more arguments which pin down the encoding and context. Consider the following scenario. I write down the number 10 on a piece of paper, and describe it as "the number on the paper." Simultaneously, someone in China writes down the number 11 on a piece of paper and also describes it as "the number on the paper." As you can see, the same description can refer to different numbers depending on the context in which the description is given. This is why $P(n)$ in not well-defined.

2 Stable Marriage

Consider a stable matching M on n boys and n girls. Suppose that Alice and Bob are married in M. Now suppose that Alice and Bob move to Canada. Then we are left with a matching, say L, on $n-1$ boys and $n-1$ girls. Is L stable?

In fact, L is stable. Suppose it is not. Then there exist some couples (b_1, g_1) and (b_2, g_2) where b_1 likes g_2 more than g_1, and g_2 likes b_1 more than b_2. But these same two couples exist in M, and with the same preferences, so b_1 and g_2 would be a rouge couple in M as well. Since M is stable, this is a contradiction.

Now consider two (not necessarily stable) matchings M and M' on the same set of people. Let $M \bigcup M'$ be the configuration in which each girl is married to the better of her two partners in M and M'. Is $M \bigcup M'$ always a matching? Recall that in a matching, each boy is married to exactly one girl, and each girl is married to exactly on boy.

Unfortunately, $M \bigcup M'$ is not guaranteed to be a matching. Suppose M contains the couples (b_1, g_1) and (b_2, g_2), and that M' contains the couples (b_2, g_1) and (b_1, g_2). Suppose that both g_1 and g_2 prefer b_1 over b_2. Then in $M \bigcup M'$, we have the pairings (b_1, g_1) and (b_1, g_2). Since b_1 is married to two girls, $M \bigcup M'$ is not a matching.

Finally, consider the situation where there are $n+1$ boys and n girls. Does the traditional marriage algorithm produce a stable pairing between n of the boys and the n girls?

In order to see that TMA does indeed produce a stable pairing, consider a virtual girl g_{n+1}. Let this girl have an arbitrary preference list, and for each boy, add this girl to the end of their preference lists. The addition of this virtual girl simulates the running of TMA on the $n+1$ boys and n girls. Now TMA produces a stable matching on this new configuration, so it does for the original. This same procedure can be used for any k boys and m girls, by introducing virtual people into the problem.

3 Cake Cutting

Consider the following cake cutting algorithm for $n = 3$:

1. A cuts the cake into three equal pieces.
2. B trims the largest piece so it is equal to the second largest piece, throwing away the trimmings.
3. C chooses the largest piece.
4. B chooses between the two remaining pieces.
5. A takes the last piece.

Is this algorithm fair? Obviously not, since C can assign the trimmings a value of 1, in which case he ends up with none of the cake. How about in a weaker sense of fair, call it *fair′*, where each person gets at least $\frac{1}{3}$ of the *remaining* cake? It is not even fair′, if A values the trimmings and ends up with a trimmed piece. (The trimmed piece would have value $< \frac{1}{3}$ of the original, while the other two would be exactly $\frac{1}{3}$ of the original, resulting in the trimmed piece being less than $\frac{1}{3}$ of the remaining.) However, it is fair′ for B and C. It is for C since C gets to pick first. It is for B since at least two of the pieces are equally the largest, and since he gets to pick second, he will get one of those.

Now the above algorithm can be made fair′ for A if we can somehow guarantee that A gets an untrimmed piece. This can be done by forcing B to take the trimmed piece if C does not pick it. Now it is still fair′ for C by the same reasoning as above. It is fair′ for A since she gets an untrimmed piece, worth $\frac{1}{3}$ of the original cake, so it must be at least $\frac{1}{3}$ of the remaining cake. It is fair′ for B, since the trimmed piece is one of the two largest pieces. Thus the modified algorithm is fair′.

In order to modify the algorithm to be fair, we need to decide what to do with the trimmings. Now it is already fair for A, since she gets an untrimmed piece. So only B and C need to participate in the partitioning of the trimmings. The obvious choice is to use cut-and-choose. So consider a new algorithm:

1. A cuts the cake into three equal pieces.
2. B trims the largest piece so it is equal to the second largest piece.
3. C chooses the largest piece.
4. B takes the trimmed piece if it is still available, or chooses between the two remaining pieces if not.
5. A takes the last piece.
6. B and C cut-and-choose the trimmings.

Is this new algorithm fair? As we said before, it is fair for A. Now consider B. Suppose he assigns a value of x_1 to the largest piece A cuts, x_2 to the second largest, and x_3 to the smallest, where $x_1 + x_2 + x_3 = 1$. It is easy to see that $x_1 \geq \frac{1}{3}$, and $x_3 \leq \frac{1}{3}$. Then the trimmings have value $x_1 - x_2$. So B gets at least $x_2 + \frac{x_1 - x_2}{2} = \frac{x_1 + x_2}{2} = \frac{1 - x_3}{2} \geq \frac{1 - 1/3}{2} = \frac{1}{3}$ of the cake. Thus it is fair for B. It is also fair for C, but I leave the proof as an exercise.

Is the algorithm envy-free? Recall that *envy-free* means that each person does the best according to his own measure. Unfortunately, it is not envy-free for A, since one of B and C get an untrimmed piece, worth $\frac{1}{3}$, plus some of the trimmings, for a total of more than the $\frac{1}{3}$ that A got.

By an extension of the above reasoning, we see that it is impossible for the algorithm to be envy-free without giving A some of the trimmings. However, we can see that no matter how much of the trimmings the person who got the trimmed piece gets, A will not envy him. This is because that person can get at most the untrimmed piece plus all the trimmings, for exactly $\frac{1}{3}$ of the cake according to A's measure, the same amount that A got. Thus I propose the following algorithm:

1. A cuts the cake into three equal pieces.
2. B trims the largest piece so it is equal to the second largest piece.
3. C chooses the largest piece.
4. B takes the trimmed piece if it is still available, or chooses between the two remaining pieces if not.
5. A takes the last piece.
6. Divide the trimmings as follows. Let T be the person who took the trimmed piece, and U be the one between B and C who took the untrimmed piece.
   (a) U cuts the trimmings into three equal pieces.
   (b) T takes one piece.
   (c) A takes a piece.
   (d) U takes the remaining piece.

It is easy to see that this algorithm is envy-free for A. We know that A cannot envy T. A also does not envy U, since she gets the same amount of the trimmed cake as U and more of the trimmings.

Now the algorithm is also envy-free for T. Consider when B gets the trimmed piece. This piece is one of the two largest pieces in the trimmed cake, so B gets at least as much of the trimmed cake as A or C. Similarly, if C is T, the he gets to choose first out of the trimmed cake, so he gets the largest piece in the trimmed cake. Since T gets the largest piece of the trimmed cake and gets the best of the trimmings, he does better than A or U.

Finally, the algorithm is envy-free for U. By a similar argument as that for T, U gets at least as much of the trimmed cake as the other two. Now since U gets to cut the trimmings, he gets the same amount of trimmings as A and T. Thus he gets at least as much of the total cake as A or T.

Unfortunately, there is no known generalization of this $n = 3$ envy-free algorithm for higher n. A complicated envy-free algorithm for $n = 4$ is known, but no envy-free algorithm for $n \geq 5$ has been discovered.