Sum and Product Puzzles

Don
Mathcamp 2016

Week 1

1 The Impossible Puzzle

The sum and product puzzle, written by Hans Freudenthal in 1969 and titled "The Impossible Puzzle" by Martin Gardner, went as follows:

Puzzle 1.1. Sam and Pesto are both Mathcampers (and thus perfect logicians). I think of two numbers, X and Y , such that $1 < X < Y$, and $X + Y < 100$. I tell Sam their sum, $X + Y$, and tell Pesto their product, $X \cdot Y$. They then have the following exchange:
P: "I do not know the values of X and Y."
S: "I knew that you did not know X and Y."
P: "Oh, now I know X and Y."
S: "Oh, now I do too."

In general, we assume that if someone knows the values of secret numbers, they say so, and if a statement begins with "oh," what follows was dependent on the previous statement.

The generalized impossible puzzle, $IMP^{\neq}(m, N)$ is as follows:

Puzzle 1.2. Sam and Pesto are both Mathcampers (and thus perfect logicians). I think of two numbers, X and Y , such that $m < X < Y$, and $X + Y < N$. I tell Sam their sum, $X + Y$, and tell Pesto their product, $X \cdot Y$. They then have the following exchange:
P: "I do not know the values of X and Y."
S: "I knew that you did not know X and Y."
P: "Oh, now I know X and Y."
S: "Oh, now I do too."

We call the same puzzle, except with $X \leq Y$, $IMP^{=}(m, N)$.

2 Variables

The impossible puzzles above take the original sum and product puzzle, and change three possible variables: the lower bound on X and Y, the upper bound on $X + Y$, and whether or not we allow $X = Y$. We also want to eventually consider changes to these following variables:

1. Which functions are used to relate the numbers our participants know.
2. The number of statements in the conversation.
3. The nature of the statements made.

The last of these will always need to be explicitly stated when defining a puzzle, but eventually we'll see puzzles that have the functions used and/or the number of statements as input variables.

3 Exercises

Exercise 3.1. If I know both the sum and product of two numbers X and Y, can I recover X and Y themselves?

Exercise 3.2. Find all solutions to $IMP^{=}(0,12)$ and $IMP^{\neq}(0,12)$

Exercise 3.3. The Consecutive Integer Game

You're playing a game against one other person, in which you each get to see one side of a card. The two sides are, in some order, a positive integer N and its successor, $N+1$. I ask you in turn whether you want to veto this round; you can say "yes", and end the round with no winner, "I don't know" and I'll ask you again after asking your opponent, or "no" - if both players say "no" then the player seeing the higher number wins. What happens in this game?

4 Denniston's Algorithm

In the course of our resolutions of $IMP^{=}(0,12)$ and $IMP^{\neq}(0,12)$, we developed an algorithm originally due to Denniston, that proceeds as follows:

1. Create a table where the columns are labeled with every possible sum, and the rows are labeled with every possible product.
2. Cross out entries that correspond to incompatible sums and products.
3. For statement 1, fill entries contradicted by that statement with a 1.
4. Repeat step 3 for each statement in order, filling the boxes they contradict with their number.
5. All remaining open entries are potential solutions to the puzzle.

If there are multiple solutions at the end of the algorithm, that means there were multiple sets of numbers compatible with all of the statements made; so both Sam and Pesto know the numbers, but we don't. If there is only one solution, then we, too, know it, and if there are no open entries, the puzzle is actually "impossible."

Further, we were able to come up with succinct rules for how to manipulate the table when presented with several basic statement types. The following rules will assume the speaker is Pesto; if it is Sam, simply switch "row" and "column."

1. "I don't know the numbers" — Eliminate all entries that are the only entry in their row.
2. "I know the numbers" — Eliminate all entries that are not the only entry in their row.
3. "i knew your previous statement" — Eliminate all entries in rows containing the number of the previous statement.
4. "i didn't know your previous statement" — Eliminate all entries in rows not containing the number of the previous statement.

We'll consider other statement types in the coming days, but these four serve as a good basis for analysis.

Finally, note that, while Denniston's Algorithm is defined in terms of a puzzle using the functions "sum" and "product," nothing in its use actually assumes anything about the functions; we simply label the rows by the possible numbers one player could know, and the columns by the possible numbers the other player could know - so we can use Denniston's Algorithm for any two-player puzzle as we have defined them.

5 Basic Statement Puzzles

The two simplest statements our players could make are "I know" and "I don't know." Let's consider sum and product puzzles in which those are the only statements made.

Puzzle 5.1. We'll define $BASIC^*(m, K)$ as the basic statement puzzle for sum and product where $*$ is either $=$ or $\neq$ specifying whether $x = y$ is allowed, $m < x$, and there are k statements of "I don't know", counting from the first time P says it.

Theorem 5.2. *Once P has said he doesn't know at least once, there are only finitely many solutions for any given length of conversation.*

To prove this theorem, we need a couple of lemmas:

Lemma 5.3. *After P says "I don't know," there are only finitely many sums for which S could deduce the value.*

Proof: For all possible sums $n \geq 4m+7$, we have $n = (2m+2)+(n-2m-2)$ and $n = (2m+4)+(n-2m-4)$, neither of which were ruled out by P's statement, because $(2m+2)\cdot(n-2m-2) = (m+1)\cdot(2n-4m-4)$ and $(2m+4)\cdot(n-2m-4) = (m+2)\cdot(2n-4m-8)$, and n is sufficiently large that those are all distinct values.

Lemma 5.4. *If the previous statement could only provide information about G pairs (x, y) for some finite G, then the next statement can provide information about at most G pairs (x, y).*

This follows because, if you execute Denniston's algorithm, only G rows/columns could have been affected by the last statement, so the next speaker can only have new information about at most G columns/rows - and they may not even be able to convey all ofit.

Putting these lemmas together, we get the theorem above.

If you experiment with specific values of m, you might see some trends starting to take place. I put here an open conjecture:

Conjecture 5.5. *There are finitely many values of m, k with $k \geq 1$ such that $BASIC^*(m, k)$ has any solutions.*

6 Solving the Impossible Puzzle

At some point, I will write up a bit about how we solved the impossible puzzle. For now, let it suffice to say: we used the algorithm, and kept track of the non-empty rows and columns in a list, rather than creating the entire table.

7 Further Results on the Impossible Puzzle

Theorem 7.1. *The set of upper bounds N for which a given pair (x, y) is consistent with the first k statements of $IMP^*(m, N)$ is:*

1. *$k = 1$: empty or of the form $[l, \infty]$*
2. *$k = 2$: empty or of the form $[l, \infty]$*
3. *$k = 3$: empty, of the form $[l, r]$ or of the form $[l, \infty]$*
4. *$k = 4$: empty, of the form $[l, r]$ or of the form $[l, \infty]$*

Pairs (x, y) for which there is an unbounded set of upper bounds making it a valid solution are called "stable solutions"; the rest are called "phantom solutions."

Theorem 7.2. *The stable solutions of $IMP^{=}(0, N)$ are all of the form $(1, z)$ where z is composite, and for every proper divisor d of z, $d + (z-1)/d - 1$ is prime.*

Theorem 7.3. *The stable solutions of $IMP^{\neq}(0, N)$ are all of the form $(1, z)$ where z is composite, and for every proper divisor d of z other than $\sqrt{z}$, $d + (z-1)/d - 1$ is prime.*

Theorem 7.4. *If $m > 0$ and we assume the strong Goldbach conjecture (that all even numbers at least 8 are the sum of two distinct primes), the stable solutions of $IMP^{=}(m, N)$ and $IMP^{\neq}(m, N)$ are the same.*

8 Problems for Investigation

There is no value of m for which the number (be it zero, finite, or infinite) of stable solutions to $IMP^{*}(m, N)$ has been proven - it is claimed to be infinite for $m = 0$, suspected to be infinite for $m = 1$ and $m = 2$, and suspected to be zero for $m \geq 3$. These proofs might be fairly hard, but also, they have only been investigated (in published sources) for values of N up to $50,000$.

There are a number of specific values of $BASIC^{*}(m, K)$ for which the number of solutions is known, but the general claim that for basic puzzles with $K > 0$ have finitely many solutions put together remains unproven - as far as I know, no one has even made an attempt to code this for investigation.