

On the NP-Completeness of Cryptarithms

David Eppstein

Computer Science Department

Columbia University

New York, NY 10027

June 8, 2000

Cryptarithm puzzles, also known as alphametics, appear widely in recreational mathematics publications, and have also been used as an example of the efficiency of constraint propagation search techniques [5]. Here we show that solving such puzzles is an NP-complete problem.

Many cryptarithms are given by Madachy [4], including the following well-known example:

$$\begin{array}{r} \text{SEND} \\ +\text{MORE} \\ \hline \text{MONEY} \end{array}$$

The puzzle is to find a one-to-one correspondence between letters in the puzzle and decimal digits (not all of which must appear) that will make the sum correct. There is usually also a rule that numbers are expressed without leading zeros. The above example has a unique solution:

$$\begin{array}{r} 9567 \\ +1085 \\ \hline 10652 \end{array}$$

If we only consider problems with decimal numbers as above there is no possibility of any hardness result, because there are only $10!$ different assignments of digits and letters to try. Therefore we will generalize the problem slightly: the base of representation for our numbers will be given as part of the problem (expressed in unary or binary) rather than always being decimal, and we will allow the puzzle to contain arbitrarily many different letters (up to the base of representation). The decision problem for a puzzle will be to determine whether that puzzle has any solutions.

Constructing the Puzzle

Given a cryptarithm puzzle under the assumptions above, it is not too hard to see that a solution to the puzzle need only be as long as the length of the base multiplied by the number of letters in the puzzle, and that such a solution can be verified quickly. Thus cryptarithms are in NP, and it remains to show that they are complete for NP. We will do this with a reduction from 3-SAT [2]: given a 3-CNF Boolean formula, we will construct a puzzle which is solvable if and only if the formula is satisfiable.

To produce enough symbols for the letters of the resulting cryptarithm, we will use subscripted letters. Each variable and term of the Boolean formula will correspond to some contiguous set of columns of the puzzle; it won't matter in what order these sets occur, except that we reserve the rightmost three columns for the following letters:

$$\begin{array}{r} 0 p 0 \\ 0 p 0 \\ \hline 1 q 0 \end{array}$$

Here 0 and 1 should not be read as the digits 0 and 1 themselves, but as letters in the puzzle which are forced by the above sequence of columns to stand for the digits 0 and 1. That is, in any possible solution to

any puzzle that has these rightmost columns, the letter 0 above must correspond to the actual digit 0, and the letter 1 must correspond to the actual digit 1.

For each variable v_i of the formula we set aside the following columns, in which the letters v_i and \bar{v}_i represent the variable and its complement:

$$\begin{array}{r} d_i 0 \ 1 \ y_i 0 \ c_i \ y_i 0 \ b_i \ y_i 0 \ a_i 0 \\ \hline e_i 0 \ d_i \ y_i 0 \ c_i \ y_i 0 \ b_i \ y_i 0 \ a_i 0 \\ \hline \bar{v}_i 0 \ e_i \ z_i 0 \ d_i \ z_i 0 \ v_i \ z_i 0 \ b_i 0 \end{array}$$

The leftmost column ($d_i + e_i = \bar{v}_i$) will not be able to carry out because of what will be to its left; the rightmost column enforces a similar restriction on whatever is to its right. The only letters in these columns that will appear anywhere else in the puzzle are v_i and \bar{v}_i . Because $b_i = 2a_i$, and $v_i = 2b_i + 0$ or 1 depending on whether $y_i + y_i$ carries, we see that v_i must be 0 or 1 mod 4 in any possible solution to the puzzle. Similarly \bar{v}_i will be 1 or 0 mod 4. We will say that a letter v_i represents a true boolean value if it is 1 mod 4, and false if it is 0; we see from the modulus relations above that \bar{v}_i in fact represents the complement of the value that v_i represents.

There will also be a set of columns for each term ($v_a \vee v_b \vee v_c$) in our 3-CNF formula (where the v_j are either variables or their complements):

$$\begin{array}{r} u_{ab} 0 \ v_a 0 \ 1 \ r_i 0 \ g_i \ w_i 0 \ f_i 0 \\ \hline v_c 0 \ v_b 0 \ h_i \ r_i 0 \ g_i \ w_i 0 \ f_i 0 \\ \hline t_i 0 \ u_{ab} 0 \ t_i \ s_i 0 \ h_i \ x_i 0 \ g_i 0 \end{array}$$

If the same v_a and v_b appear in more than one term, we must use the same letter u_{ab} for all of those terms. The rightmost columns (those with $f_i, g_i, w_i, h_i, r_i,$ and s_i) force t_i to be one of 1, 2, or 3 mod 4 in a manner similar to the way v_i was forced to be either 0 or 1 mod 4. Then the leftmost columns force t_i to equal $v_a + v_b + v_c$, and since each of these is 0 or 1 mod 4, the fact that the sum is non-zero mod 4 forces at least one of the letters v_j to be 1 mod 4.

The above shows that, given a solution to the cryptarithm, we can turn it into an assignment to the variables of the formula in the manner discussed above, and by the restrictions forced in the columns corresponding to the terms it will be the case that each term in the formula will have at least one of its variables set to true. Thus a solution to the puzzle gives us a satisfying assignment to the formula. This will be true no matter what base we choose for our arithmetic, but to complete the NP-completeness proof we must find a base that will give us the converse, so that from a satisfying assignment to the formula we can find a solution to the puzzle. It will turn out that with the base equal to $3072n^3$, where n is the number of variables in the formula, we can in fact find a solution if a satisfying assignment exists. This number is small enough to be represented in unary, and so all that remains to be proved is that the constructed puzzle is as claimed solvable.

Solving the Puzzle

First we will restrict our attention to solutions in which the letters of the puzzle have the following values taken modulo 128:

Letter:	a	b	c	d	e	f	g	h	v, \bar{v}	u	t	p, r, w, y	q, s, x, z
Value:	2, 34, 66, 98	4, 68	1, 2, 33, 34, 65, 66, 97, 98	3, 4, 67, 68	5, 69	6, 38, 70, 102	12, 76	24, 25	8, 9	16, 17, 18	25, 26, 27	7, 71	14

We will also assume that our base is divisible by 128. Define the class of x for any letter x to be $\lfloor x/128 \rfloor$. From the column $b_i + b_i + \text{carry} = v_i$ in the terms for that variable, we can deduce that $b_i = \lfloor v_i/2 \rfloor$, so that if v_i is fixed to have a given class, the value of b_i is determined. Similarly all letters except those in the last two columns above will be fixed by the choice of v_i and \bar{v}_i . We will be unable to solve the problem if two letters take the same value, but the modulus relations make sure that this is only possible if there are two triples of variables or their complements such that the sums of the classes in each triple are the same (the only possible collisions occur when two v_i or \bar{v}_i , two u_{ab} , or two t_i have the same class, and the above covers all three cases).

We also need to assign values to the letters in the last two columns, but that is easy since each pair doesn't interfere with any other pair (by the modulus relations) and any setting of q , s , x or z will allow the corresponding sum to either carry or not carry as desired. So we need as many classes as there are such letters in the puzzle, but this turns out to be less than we will need for the variables. The only other obstacle to a solution of the puzzle is that some digits might need to be larger than the base of arithmetic. We will resolve this by setting the base to be at least three times the value assigned to the largest variable.

Thus our problem reduces to finding an assignment for the classes of the v_i and \bar{v}_i letters such that all sums in triples are distinct. If we can do that, and given the modulus relations above, it will be an easy task to assign v_i to be 8 or 9 mod 128 depending on whether or not the variable is true in our hypothetical satisfying assignment, and this will induce an assignment for all the other letters.

As a first try at generating a non-colliding assignment, we will set the classes of v_i to be powers of 2. Any sum of distinct powers of two is distinct, so this satisfies our distinct triple requirement. However, such an assignment will cause the base to be $O(4^n)$, which is too large to be written in unary (but can be written in binary). So this doesn't give us as strong a result as we would like.

To find a solution with a smaller base, we must turn to a 1959 result of Bose and Chowla. They proved that for any k there is a set of k numbers, all between 1 and k^3 , such that their sums in triples are distinct [1,3]. Thus we only need our classes to run from 1 to $(2n)^3$. If we multiply this by $128 \cdot 3$ for the values within each class and for room for the terms as described above, we see that from a satisfying assignment to the 3-CNF formula we can find a solution to the derived puzzle with a base of only $3072n^3$. Thus with this as our base the puzzle can be solved if and only if the formula is satisfiable, and so as claimed cryptarithms are NP-complete even when the base must be written in unary.

References

- [1] S.C. Bose and S. Chowla, *Report Inst. Theory of Numbers*, University of Colorado, Boulder, 1959.
- [2] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, 1979.
- [3] Richard K. Guy, *Unsolved Problems in Number Theory*, Springer-Verlag, 1981, p. 68.
- [4] Joseph S. Madachy, *Madachy's Mathematical Recreations*, Dover, 1979, pp. 178–200.
- [5] Elaine Rich, *Artificial Intelligence*, McGraw Hill, 1983, pp. 95–98.