NP-Completeness of Crytarithms: An Exposition

William Gasarch *

January 30, 2021

1 Introduction

Cryptarithms are classic puzzles involving arithmetic on words. Figure 1 gives an example.

$$\begin{array}{cccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

Figure 1: The SEND MORE MONEY Cryptarithm.

Here is how one might begin solving the cryptarithms in Figure 1:

1. A carry can be at most 1. Hence $M = 1$.

2. Since the last column is a carry and $M = 1$, the digit before that must be either 0 or 1. Since $O \neq M$, we have $O = 0$, which is convenient.

3. The E, O, N column can't contribute a carry: Assume that it did. Then since $O = 0$ we would have that $N + R$ contributes a carry, and $E + 1$ results in a carry. Then $E = 9$ and $N = 0$. But this is impossible since $O = 0$.

4. Since $M = 1$, $O = 0$ and the E, O, N column does not contribute a carry, $S = 9$.

*University of Maryland, College Park, MD 20742, gasarch@cs.umd.edu

5. To recap we have $M = 1$, $O = 0$, $S = 9$.

6. Look at the E, O, N column. Since $O = 0$, this column does not contribute a carry, and $E \neq N$ (by the rules of the puzzle), the (N, R, E) column has to contribute a carry. So we have $E + 1 \equiv N \pmod{10}$, and $E, N \notin \{0, 1, 9\}$. Hence

$$(E, N) \in \{(2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8)\}.$$

We stop here; however, notice that we may end up encountering many possibilities. If you work them through you will find that the answer is unique and is:

$$S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2.$$

Figure 2 shows how to check the answer.

$$\begin{array}{cccccc} & 9 & 5 & 6 & 7 \\ + & 1 & 0 & 8 & 5 \\ \hline 1 & 0 & 6 & 5 & 2 \end{array}$$

Figure 2: Solution to The SEND MORE MONEY Cryptarithm.

Are there puzzles like this where solving them leads to many cases? Likely yes: David Eppstein [**?**] showed

$$\text{3SAT} \leq_p \text{CRYPTARITHMS}$$

and hence CRYPTARITHMS is NPC. We will show

$$\text{MONO 1-IN-3-SAT} \leq_p \text{CRYPTARITHMS}$$

which, by Theorem **??**, also yields that CRYPTARITHMS is NPC; however, our proof is easier.

We first need to define the problem rigorously.

Def 1.1 CRYPTARITHMS is the following problem:

1. Input:

   - $B, m \in \mathcal{N}$. Let Σ be an alphabet of B letters.
   - $x_0, \ldots, x_{m-1}$. Each $x_i \in \Sigma$.
   - $y_0, \ldots, y_{m-1}$. Each $y_i \in \Sigma$.
   - $z_0, \ldots, z_m$. Each $z_i \in \Sigma$. The symbol z_m is optional.

2. Output: YES if there exists an injection of Σ into $\{0, \ldots, B-1\}$ so that the arithmetic statement in Figure 3 is true (in base B). NO otherwise.

$$\begin{array}{cccc} & x_{m-1} & \cdots & x_0 \\ + & y_{m-1} & \cdots & y_0 \\ \hline z_m & z_{m-1} & \cdots & z_0 \end{array}$$

Figure 3: Solution to CRYPTARITHMS

Theorem 1.2 CRYPTARITHMS *is* NPC.

Proof:

Given $\phi = C_1 \wedge \cdots \wedge C_k$, a formula where every literal is positive, we want to create an instance J of CRYPTARITHMS such that the following are equivalent:

- There exists an assignment that satisfies exactly one literal per clause.
- There exists a solution to J.

Let n be the number of variables in ϕ. As shown, k is the number of clauses.

We will determine the base B, and the length of the numbers m, later.

1) *Constants* We need two letters that we suggestively call 0 and 1 that we force to map to 0 and 1. We use the columns in Figure 4.

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

Figure 4: The Constants Gadget.

It is easy to see that the columns in Figure 4 force the letter "0" to have the value 0 and the letter "1" to have the value 1. (You will need to use that carries are either 0 or 1.)

To establish the constants 0 and 1 we need (1) the 4 letters $0, 1, p, q$, and (2) 3 columns.

2) *Variables* Let v be a variable in ϕ.

v is considered true if $v \equiv 1 \pmod 4$ and false if $v \equiv 0 \pmod 4$

Hence we need to ensure that $v \equiv 0 \pmod 4$ or $v \equiv 1 \pmod 4$.

This is accomplished through a string of intermediate sums:

$$\begin{aligned} b &= 2a \\ 2c &= d + C \qquad\qquad C = \text{carry}(c + c) \in \{0, 1\} \\ v &= 2b + C \\ &= 4a + C \equiv C \pmod 4 \end{aligned}$$

(Note that a, b, c, d, v are used in the puzzle, whereas C is not.)

We do not have to consider $\overline{v}$ since our formula has all positive literals.

$$\begin{array}{cccccc} 0 & b & c & 0 & a & 0 \\ 0 & b & c & 0 & a & 0 \\ \hline 0 & v & d & 0 & b & 0 \end{array}$$

Figure 5: The Variable Gadget.

Each variable v needs (1) the 5 letters a, b, c, d, v and (2) 6 columns. Since there are are n variables, we need $5n$ letters and $6n$ columns for this part.

3) *Clauses* Let C be the clause $x \vee y \vee z$ (x, y, z need not be distinct). Let x, y, z be the letters corresponding to the those variables in the cryptarithm. Since our reduction is from MONO 1-IN-3-SAT $\leq_p$ CRYPTARITHMS we need that $x + y + z \equiv 1 \pmod 4$.

We first need to have a number d that can be anything $\equiv 1 \pmod 4$. This is accomplished by the following equations:

$$\begin{aligned} b &= 2a \\ c &= 2b \\ &= 4a \\ d &= c + 1 \\ &= 4a + 1 \end{aligned}$$

We then need to have that $x + y + z = d$. We need an intermediary variable for $x + y$ that we call I (for Intermediate).

$$x + y = I$$

$$I + z = d$$

The result is Figure 6

$$\begin{array}{cccccccccccc} 0 & I & 0 & x & 0 & 1 & 0 & b & 0 & a & 0 \\ 0 & z & 0 & y & 0 & c & 0 & b & 0 & a & 0 \\ \hline 0 & d & 0 & I & 0 & d & 0 & c & 0 & b & 0 \end{array}$$

Figure 6: The clause gadget.

Each clause C needs (1) the 5 letters a, b, c, d, I and (2) 11 columns. Since there are k clauses, we need $5k$ letters and $11k$ columns.

The construction is completed.

The final instance of CRYPTARITHMS uses $5n + 5k + 4$ letters and $6n + 11k + 3$ columns. However, we cannot just take $B = 5n + 5k + 4$. We need enough numbers so that, for example, (looking at the clause gadget) we don't have $b + b$ is the same as $x + y$. We revisit the issue of B soon.

Clearly, a solution to the cryptarithm J gives a solution to the MONO 1-IN-3-SAT problem ϕ. The other direction is less clear. Assume we have a solution to the MONO 1-IN-3-SAT problem ϕ. This assigns T or F to each of the variables $v_1, \ldots, v_n$. We translate this to an assignment of the letters $v_1, \ldots, v_n$ to numbers. We do this inductively. Assume letters $v_1, \ldots, v_{i-1}$ and some of the other letters have been assigned.

1. If variable v_i is T then we will assign letter v_i to a number $\equiv 1 \pmod 4$.

2. Assign to the letter v_i the least number that has the right congruence mod 4, has not been assigned to any other letter, and does not cause any letter to be assigned to a an already-used number. This arises (1) with the variable gadgets since once you assign v you need to assign a, b, c, d, and (2) with any clause gadget that contains v where the other variables in it have been assigned.

We leave it to the reader to determine how large B must be to accommodates all these numbers.

- B will be large enough so that many numbers will not be used. Hence there will be letters that are not used mapping to numbers we do not use. This is not a problem.
- B will be bounded by a polynomial in k, n. Hence CRYPTARITHMS is strongly NPC.

∎

Exercise 1 Do a direct reduction 3SAT $\leq_p$ CRYPTARITHMS.

Exercise 2

1. Write an algorithm for this problem and analyze its run time.
2. How fast is your algorithm when B is a constant? (It should be polynomial time.)