Applied Logic Weekly Assignment Report

Bohdan Tymofieienko B.S.
b.tymofieienko@student.fontys.nl
4132645

Mikolaj Hilgert M.H.
m.hilgert@student.fontys.nl
4158385

1 Week i. Alphametic puzzles. Inverse matrix.

1.1 Alphametic puzzle

The point is to use the digits 0,1,2,3,4,5,6,7,8,9 to solve puzzle where words are put together into an arithmetic formula such that digits can be substituted for the letters to make the formula true.

The rules for an alphanumeric puzzles are, that no letter can have more than one value associated with [eg, if N=5, then N=5 for all N] and no value can be associated with more than one letter [eg, both Z and N can't be 5]. *(Manan Shah, 2017)*

For the sake of clarity we chose a 'shorter' puzzle, as the general approach using *Z3* does not change, rather more code has to be written at the cost of legibility. Consequently, the puzzle we chose was:

I + DID = TOO

To solve this puzzle with *Z3*, an essential step one must take, is to take the rules of the puzzle and transform them into logical assertions. The first step is to define our alphabet of letters used for the problem, as well as two constants used for sum comparison.

```
(declare-const I Int)
(declare-const D Int)
(declare-const T Int)
(declare-const O Int)

(declare-const Input Int)
(declare-const Output Int)
```

As such, per the rules, a digit can only be assigned to one letter. We thus make the assertion that all values of letters must be **distinct** (Non repeating).

```
(assert (distinct I D T O))
```

Then, we must add constraints to the values so that they may only line within the bounds [0,9].

```
(assert (and (>= I 0) (<= I 9)))
(assert (and (>= D 0) (<= D 9)))
(assert (and (>= T 0) (<= T 9)))
(assert (and (>= O 0) (<= O 9)))
```

Now that the given conditions are set. We can begin constructing the actual words. Therefore, we can rewrite the addition into the following equation:

$$I + (100D + 10I + D) = 100T + 10O + O \quad (1)$$

The fact that the two have to be equal, means we can make an assertion within *Z3*.

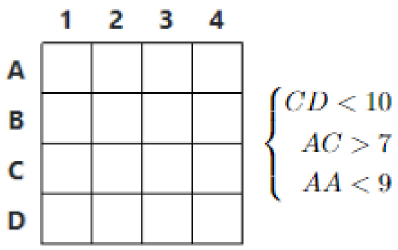
```
(assert(= Input (+ (* 1 I)
(+ (* 100 D) (* 10 I) (* 1 D)))))

(assert(= Output (+ (* 100 T)
(* 10 O) (+ (* 100 D) (* 1 O)))))

(assert (= Output Input))

(check-sat)
(get-model)
```

All of these steps are important, as essentially what we tell *Z3* to do, is to go through and find values for I, D, T and O, for which all of the conditions are satisfied. Which will result in the correct addition. In this case finding that D=1 I=9 O=0 T=2. When substituted into equation (1), you get a result of **9+191 = 200**.

1.2 Logic grid puzzle

The approach is very similar to alphametic puzzles. The target is to set the constraints in a way SMT may find a suitable values to satisfy the conditions.

According to the rules of the game, there is a set of inequalities and equations that has to be satisfied. There also exist a set of variables A and a set of possible values S. Essential condition is one-to-one mapping, meaning only distinct values can be assigned to distinct elements. Similarly in the previous puzzle, you must find an arrangement of values which will satisfy the conditions.



Essentially, the conditions of the game could be interpreted in form of logical statements. As shown below:

```
(assert(=Condition1(<(* C D)10)))
(assert(=Condition2(>(* A C)7)))
(assert(=Condition3(<(* A A)9)))
```

Additionally, it is important to add bounds to possible values. That is similar to who it is done in Alphametic puzzle solution.
In this case $1 \leq x \leq 4$ applies.

1.3 Inverse matrix

In essence, matrix is an $m \times n$ array of scalars, namely entries [1].
(Org, Kjell, Definition of a Matrix)

$$\begin{bmatrix} a_1 1 & a_1 2 & \dots & a_1 n \\ a_2 1 & a_2 2 & \dots & a_2 n \\ \vdots & \vdots & \ddots & \vdots \\ a_m 1 & a_m 2 & \dots & a_m n \end{bmatrix}$$

Figure 1: $n \times m$ matrix

An $n \times n$ square matrix A is called invertible if there exist an $n \times n$ matrix B such that,

$$AB = BA = I \tag{2}$$

where I denotes an $n \times n$ identity matrix.

To compute an inverse matrix, one way is to use the determinant,

$$\det A = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{3}$$

Multiplying a reciprocal of a determinant by a matrix A with a and d places and c, b multiplied by -1 will result in an inverse matrix.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{detA} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{4}$$

By conditions of the assignment only invertible matrices with inverse matrix consisting of integer entries are allowed for input. Matrix is clearly not invertible when determinant is 0. Thus, we have to declare next:

```
(not (= Determinant 0))
```

If entry divided by a determinant has a reminder, matrix cannot be accepted for input. Adding next verification, we restrict such an input and statement happens to be unsatisfiable.

```
(=(mod Entry Determinant) 0)
```

Using $(get - valueEntry)$ operations we deduce the inverse matrix from Z3.

References

Manan Shah 2017. *Alphanumeric Puzzle 1 — Math Misery? Mathmisery.com.* http://mathmisery.com/wp/2017/01/12/alphanumeric-puzzle-1/.

Definition of Matrix *Org, Kjell At Ieee Dot. Definition of Matrix* https://chortle.ccsu.edu/vectorlessons/vmch13/vmch13$_2.html$.